NDIA

National Defense Industrial Association

DEFENSE TECHNICAL INFORMATION CENTER

Files are in Adobe format.
Download the newest version from Adobe.

2011 INFORMATION SYSTEMS SUMMIT II

"What's All This Stuff About, Anyway?"

Baltimore, MD

4 – 6 April 2011

Agenda

Monday, April 4, 2011

THE AGILE PROCESS: Iterative Incremental Lifecycle; Lightweight Processes Including Scrum and Tools

- Mr. Jeff Payne, CEO and Founder, Coveros, Inc**.**

AGILE REQUIREMENTS: User Stories, Non-functional Requirements, The Product Backlog

- Dr. Ahmed Sidky, Executive Vice President, Santeon Group

AGILE DEVELOPMENT: Test First Development, Pair Programming, Automated Unit Testing, Continuous Integration, Refactoring

- Mr. Jeff Payne, CEO and Founder, Coveros, Inc.

AGILE TESTING: Agile Testing Quadrants, Automation, Exploratory Testing, Requirements Expressed As Tests

- Dr. Ahmed Sidky, Executive Vice President, Santeon Group

Tuesday, April 5, 2011

WELCOME REMARKS

- Dr. Steve Kimmel, Chairman, NDIA C4ISR Division; Senior Vice President, Alion Science & Technology

THE DEMAND: INFORMATION SHARING — THE USER PERSPECTIVE OF WARFIGHTER NEEDS AND EXPECTATIONS — OPERATIONALIZING THE CLOUD

- COL Timothy P. Hill, USA, Director, Futures, Army Intelligence and Security Command (INSCOM)

EARNED VALUE MANAGEMENT + AGILE DEVELOPMENT = IT PROGRAM SUCCESS

- Mr. Glen Alleman, Principle of Practices, Lewis & Fowler

AGILE PLANNING AND ESTIMATION

- Dr. Ahmed Sidky, Executive Vice President, Santeon Group

THE ROLE OF THE PRODUCT OWNER

- Mr. Mike Cox, Senior Consultant, Net Objectives

MANAGING THE AGILE TEAM

- Dr. Ahmed Sidky, Executive Vice President, Santeon Group

CREATING AN EFFECTIVE BACKLOG

- Dr. Ahmed Sidky, Executive Vice President, Santeon Group

SECURE AGILE DEVELOPMENT

- Mr. Jeff Payne, CEO and Founder, Coveros, Inc.

LEAN AND KANBAN

- Mr. Mike Cox, Senior Consultant, Net Objectives

Wednesday, April 6, 2011

WELCOME REMARKS

- Dr. Steve Kimmel, Chairman, NDIA C4ISR Division; Senior Vice President, Alion Science & Technology

NDIA
National Defense Industrial Association

PROMOTING NATIONAL SECURITY SINCE 1919

INFORMATION SYSTEMS SUMMIT II

SPONSORED BY THE NDIA C4ISR DIVISION

The Summit will feature industry subject matter experts whose tutorial and track session presentations will address the Summit's theme — "What's All This Agile Stuff About, Anyway?"

SUMMIT AGENDA

HYATT REGENCY BALTIMORE ▸ BALTIMORE, MD

APRIL 4-6, 2011

WWW.NDIA.ORG/MEETINGS/1750

EVENT #1750

THE VALUE PROPOSITION

Agile development and test will improve DoD's acquisition of IT applications leveraging cloud computing and service-oriented architectures used by information -sharing applications such as collaboration (strategic and tactical intelligence) analysis, ISR sensor "fusion" processing, coalition and joint tactical operations, logistics and sustainment support, transportation functions, geospatial and decision support applications, medical or health care record sharing, etc.).

The new paradigm is significantly different and should not be confused with today's spiral development process. Agile sprints are comprised of (smaller line-of-code) projects, month-not-years time driven release commitments with integrated development, operational, interoperability and user-acceptance testing.

Once implemented, DoD user-approved requirements tailored for each sprint (vice large requirement comprised major programs of record) will create a new "partnership" relationship amongst government and industry users, developers and testers.

THE PURPOSE

The Information Systems Summit II is being convened as a forum to learn and, subsequently, adapt from world-class agile commercial practitioners applicable procedures to rapidly acquire DoD information technology embellished solutions to meet warfighter needs.

BACKGROUND

Agile software IT application practice is based on an interactive and incremental development of combined software development methodologies where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams: requirements, contracts, developers, testers, and users. Nearly a decade old, this methodology has taken root in the commercial practice of Google, Microsoft, Apple and many other notable commercial entities.

CONFERENCE GOAL

Understand and accelerate the adaption of the Manifesto for Agile Software Development within DoD.

MANIFESTO FOR AGILE SOFTWARE DEVELOPMENT

Uncover better ways of developing software by doing it and helping others do it.

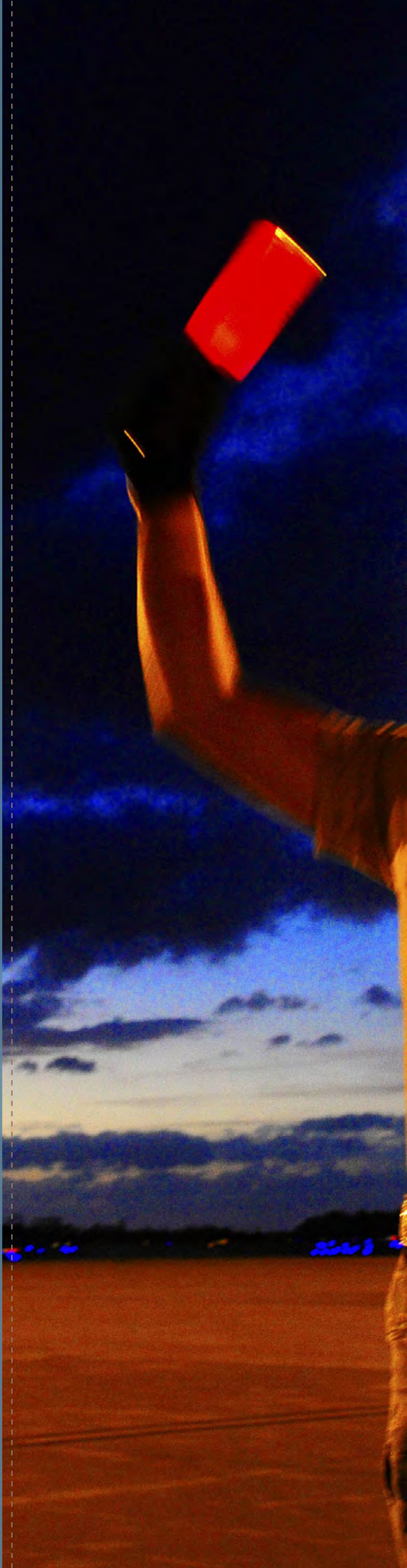
Through this work, come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, value the items on the left more.

CONFERENCE ATTIRE

Conference attire is business for civilians and uniform of the day for military. In addition, your identification badge, received upon conference check-in, must be worn at all times.



FEATURED SPEAKER PROFILES - *Listed in Order of Appearance*

- Mr. Jeff Payne, *CEO and Founder, Coveros, Inc.*

Jeff Payne is CEO and founder of Coveros, Inc., a consulting company that uses agile methods to accelerate the delivery of secure, reliable software. Prior to Coveros, Jeff was co-founder, chairman of the board, and CEO of Cigital, Inc., a market leader in application security and software quality solutions. A recognized software expert, he speaks to companies nationwide about the business risks of software failure. Jeff has been a keynote and featured speaker at CIO and business technology conferences and testifies before Congress on issues of national importance, including intellectual property rights, cyber terrorism, and software quality.

- Dr. Ahmed Sidky, *Executive Vice President, Santeon Group*

In addition to being co-author of a top-rated agile adoption book, *Becoming Agile in an Imperfect World*, Ahmed Sidky is the executive vice president at Santeon Group responsible for software delivery and agile services. He has gained popularity and respect in the agile community as a proponent of a pragmatic approach for organizations attempting to adopt agile. Ahmed is often called Dr. Agile because of his free online agile readiness assessment tool, Doctor Agile. He is a frequent speaker at national and international agile conferences. Ahmed helps guide both small and large organizations during their transition to agile software development, and enjoys coaching and educating agile teams worldwide. You can reach Ahmed at asidky@santeon.com.

- Mr. Nate Oster, *Agile Player-Coach and Founder, CodeSquads LLC*

Nate Oster is an agile player-coach and founder of CodeSquads LLC, where he helps clients adopt agile methods. Nate builds high-performance teams that emphasize continuously measuring progress with tested features, exercising all skills in parallel, and frequently delivering high-quality software that delights customers. Nate inspires adopters with hands-on mentoring and simulations that provide a safe learning environment for new ideas. He promotes testing as a serious technical discipline and is frequently consulted as an expert in test automation and system performance engineering. You can contact Nate at NateOster@CodeSquads.com.

- Mr. Don Boian, *Technical Director, Operations, USCYBERCOM*

Mr. Boian is currently the Technical Director for the Chief of Operations (J3) of the USCYBERCOM. As the Technical Director for the J3, he is responsible for providing technical and operational leadership to USCYBERCOM personnel and operations. He is also responsible for establishing and maintaining key partnerships with Department of Defense Science and Technology Community and the US Intelligence Community. Mr. Boian's prior positions include: Signals Intelligence Directorate Cyber Lead (Apr. 2009 - Nov 2009); Director of Operations, Tailored Access Operations Group (TAO) (Oct. 2007 - Mar. 2009); Chief, Remote Operations Center (ROC), TAO (Feb 2005 - Sep. 2007); Mission Director, Remote Operations Center, TAO (Apr. 2002 - Feb. 2005); Division Chief, Infrastructure and Data Networks Division (IDND), Data Network Technologies Office (DNT), TAO (Feb. 2001 - Apr. 2002); Division Chief, System Integration & Infrastructure Division (SliD), Remote Network Solutions Office (RNS), Tailored Access Solutions Group (TASG) (Jan. 2000 - Feb. 2001); Branch Chief, System Engineering, Applied Techniques Branch, Data Communications Division (K734, K731) (Jun. 1996 - Jan. 2000); Project Engineer / JOSHUA Team Leader (K153/K73) (Sep. 1994 - Jun. 1996); Information System Security Organization (ISSO) Engineer/Project Manager (Jul. 1987 - Sep1994). Mr. Boian's significant awards include: Dr. Louis Tordella Award (DIRNSA/UK-GCHQ) (Mar. 2003); Meritorious Civilian Service Award (DDO) (Sep. 2000); National Intelligence Meritorious Unit Citation (DCI) (Sep. 1997); National Intelligence Meritorious Unit Citation (DDT) (Sep. 1997); Joint Meritorious Unit Award (May 1997). Don received his Bachelor of Science Electrical Engineering with Computer Option from The Ohio State University in June of 1987 and his Master of Science Electrical Engineering Telecommunications from Johns Hopkins University in December of 1994. Mr. Boian currently resides in Woodbine, MD with his wife, Kim, and has a daughter, Elizabeth, who attends the University of Findlay in Ohio.

FEATURED SPEAKER PROFILES - *Listed in Order of Appearance*

- Mr. Sanjiv Augustine, *President, LitheSpeed*

President of LitheSpeed and an industry-leading agile expert, Sanjiv Augustine has, for more than ten years, assisted leading clients — Nationwide Insurance, Capital One, CNBC, T. Rowe Price and StreamSage — adopt agile methods. He is the author of several publications including The Lean-Agile PMO and the book, *Managing Agile Projects*. Sanjiv is the founder of the Yahoo! Agile Project Management group, co-founder of the Agile Project Leadership Network, and member of the Project Management Institute Agile Community of Practice. As an in-the-trenches practitioner, he has personally managed agile projects varying in size from five to more than one hundred people, trained thousands of agile practitioners via public classes and conference presentations, and coached numerous project teams.

- COL Timothy P. Hill, USA, *Director, Futures, Army Intelligence and Security Command (INSCOM)*

COL Timothy P. Hill was commissioned as a Military Intelligence Officer and awarded a Bachelor of Science degree upon graduation from the United States Military Academy, West Point in 1983. After commissioning, he was assigned to the 5th infantry Division (Mechanized) at Fort Polk, Louisiana. There he served in a myriad of tactical positions over six years to include: S2/Intelligence Officer for I-55th Air Defense Artillery Battalion, Platoon Leader, Company Executive Officer, and Battalion S-1 in the 105th Military Intelligence Battalion. In 1988 he assumed command of A Company, 105th Military Intelligence Battalion. Upon completion of command in 1990, he attended the Naval Postgraduate School in Monterey, California, where he received a Masters in Science Degree in Electronic Warfare Systems Engineering. He was then assigned to the advanced Technology and Concepts Division of Combat Development at Fort Huachuca, Arizona. After attending Command and General Staff College in 1995, COL Hill served as the G2, XVIII Airborne Corps Artillery. He then served as the Executive Officer, 319th Military Intelligence Battalion (Operations) (Airborne) and he completed this tour as the Chief of the XVIII Airborne Corps Analysis and Control Element. In 1998, he was assigned to the 704th Military Intelligence Brigade at Fort Meade, Maryland, as the Executive Assistant for the Director of Military Operations at the National Security Agency (NSA). COL Hill commanded the 279th Base Support Battalion in Bamberg, Germany from November 2000 until July 2003. After attending the National War College in 2003, where he received a Masters of Science degree in National Security Strategy, he served in NSA's National Cryptologic Office in the Pentagon supporting DOD wide customers. COL Hill deployed to Iraq in May 2005 and served as the Chief of the Intelligence Transition team assisting the Iraqi Ministry of Defense (MOD) Intelligence service. He completed this tour serving as the Director of the Strategic Intelligence Engagement Office. He is now serving as the Director of the INSCOM Futures Directorate. COL Hill's military schools include: Military Intelligence Officer Basic and Advanced courses, CAS3, Command and General Staff College, Armed Forces Staff College, National War College, Airborne and Jumpmaster schools. His decorations include: the Bronze Star Medal, Defense Meritorious Service Medal, Meritorious Service Metal with two Oak Leaf Clusters, Joint Service Commendation Medal, Army Commendation Medal, Army Achievement Medal, and the Senior Parachutist Badge.

- Mr. Glen Alleman, *Principle of Practices, Lewis & Fowler*

Glen B. Alleman leads the Program Planning and Controls practice for Lewis & Fowler. In this position, Glen brings his 30 years experience in program management, systems engineering, software development, and general management to bear on the problems of performance based program management. Mr. Alleman's experience ranges from real time process control in a variety of technical domains to product development management and Program Management in a variety of firms including Logicon, TRW, CH2M Hill, SM&A, and several consulting firms before joining Lewis & Fowler. Mr. Alleman's teaching experience includes university level course in mathematics, physics, and computer science. Currently, Mr. Alleman is the Principle of Practices at Lewis & Fowler and the developer of the Deliverables Based Planning® method Lewis & Fowler applies to its Aerospace, Defense, and Enterprise IT engagements. This method is applied from proposal activities through program execution focusing on IMP/IMS, programmatic risk, Technical Performance Measures, CAM and PP&C mentoring and training, process improvement, DCMA Validation, and increasing the probability of success for mission critical programs.

- Mr. Mike Cox, *Senior Consultant, Net Objectives*

Michael Cox is senior consultant for Net Objectives. He was previously with Lockheed Martin where he held a series of increasingly responsible positions in operations and program management, performing diverse functional and programmatic roles that spanned disciplines from rocket propulsion, structural engineering, and software integration to leadership development, program performance, and lean/agile implementations and process improvement. Mike's roles have included a staff assignment at the corporate headquarters Operating Excellence office as an engineering subject matter expert in lean process improvement specializing in lean/agile software development. He holds a BS in Aerospace Engineering from the University of Virginia.

FEATURED SPEAKER PROFILES - *Listed in Order of Appearance*

- Honorable Frank Kendall, *Principal Deputy Under Secretary of Defense, AT&L, OSD*

Mr. Frank Kendall was sworn in as Principal Deputy Under Secretary of Defense for Acquisition, Technology, and Logistics (PDUSD(AT&L)) on March 5, 2010. In his role as PDUSD(AT&L), Mr. Kendall is authorized to act for and provide assistance to the Under Secretary of Defense for Acquisition, Technology & Logistics (USD(AT&L)). He also advises and assists the USD(AT&L) in providing staff advice and assistance to the Secretary of Defense on the acquisition system; research and development; modeling and simulation; systems engineering; advanced technology and developmental test and evaluation. Within government, Mr. Kendall held the position of Director of Tactical Warfare Programs in the Office of the Secretary of Defense and the position of Assistant Deputy Under Secretary of Defense for Strategic Defense Systems. Mr. Kendall was also Vice President of Engineering for Raytheon Company. Mr. Kendall also spent ten years on active duty with the Army serving in Germany, teaching Engineering at West Point, and holding research and development positions. He is a Distinguished Graduate of the U.S. Military Academy at West Point and he holds a Masters Degree in Aerospace Engineering from California Institute of Technology, a Master of Business Administration degree from C.W. Post Center of Long Island University, and a Juris Doctoris from Georgetown University Law Center.

- Mr. David M. Wennergren, *Assistant Deputy Chief Management Officer, Department of Defense*

Mr. David M. Wennergren serves as the Department of Defense Assistant Deputy Chief Management Officer, where he is the principal deputy to the DoD Deputy Chief Management Officer and champions the Department's efforts to better synchronize, integrate, coordinate and improve DoD business operations. His efforts focus on achieving greater effectiveness, increased efficiency and improved performance in the Department's enterprise policies, processes, and systems. He also serves as the Director of the DoD Business Transformation Agency. Prior to his current assignment, Mr. Wennergren served as Deputy Assistant Secretary of Defense for Information Management, Integration and Technology/Deputy Chief Information Officer, where he led the creation and implementation of a unified information management and technology vision for the Department. In addition to these duties, Mr. Wennergren served for five years as the Vice Chair of the U.S. Government's Federal CIO Council, as well as serving as the Chair of the Department of Defense Identity Protection and Management Senior Coordinating Group and Chair of the Committee for National Security Systems. Prior to joining the staff of the Secretary of Defense, Mr. Wennergren served for four years as the Department of the Navy Chief Information Officer (DON CIO), during which time he also served as the Department of the Navy's Critical Infrastructure Assurance Officer. Prior to becoming the DON CIO, he served for four years as the DON Deputy CIO for Enterprise Integration and Security. Past assignments also included, the Head, Plans and Policy Branch within the Shore Installation Management Division, Office of the Deputy Chief of Naval Operations (Logistics), the Economic Support Team Leader on the Department of the Navy's Base Structure Analysis Team (BSAT) during the Navy's Base Realignment and Closure (BRAC) process for BRAC-93 and BRAC-95, Commercial Activities Program planning and review in the Office of the Deputy Chief of Naval Operations (Logistics), participating in the Navy's BRAC-91 process, and working as a management analyst at both the Naval Industrial Resources Support Activity and the Naval Air Technical Services Facility. Mr. Wennergren received his B.A. in Communications/Public Relations from Mansfield State University. He was a recipient of a Secretary of the Navy Civilian Fellowship in Financial Management, culminating in a Master of Public Policy (MPP) in Public Sector Financial Management from the University of Maryland's School of Public Affairs. He has received the Department of Defense Distinguished Civilian Service Award, the Department of the Navy Distinguished, Superior and Meritorious Civilian Service Awards, the Secretary of Defense Meritorious Civilian Service Award, and the Office of the Secretary of Defense Exceptional Civilian Service Award. Other honors include being selected as the TechAmerica Terman Award 2010 Government Technology Executive of the Year, the Federal CIO Council 2008 Azimuth Award winner, the Government Computer News 2005 Defense Executive of the Year, the 2006 John J. Franke Jr. Award from the American Council for Technology, the Federal Computer Week 2006 Eagle Award, three Federal Computer Week Fed 100 Awards, the Computerworld Premiere 100 Award, and the 2008 General James M. Rockwell AFCEAN of the Year. He is also honored to have worked in two organizations that were awarded the Department of the Navy Meritorious Unit Commendation.

FEATURED SPEAKER PROFILES - *Listed in Order of Appearance*

- Mr. Rob Carey, *Deputy Chief Information Officer, Department of Defense*

Mr. Robert J. Carey serves as the Deputy Assistant Secretary of Defense (Information Management, Integration and Technology) / Department of Defense Deputy Chief Information Officer. Selected to this position after a brief tour as Director of Strategy and Policy for the US TENTH FLEET / FLEET CYBER COMMAND his principle roles will be to help lead the consolidation of Defense information technology enterprise as well as align, strengthen and manage the office of the DoD Chief Information Officer to have it better serve the Department's mission and help lead the IT workforce into the 21st century. From November 2006 to September 2010 he served as served as the fifth Department of the Navy (DON) Chief Information Officer (CIO) where he championed transformation, enterprise services, the use of the internet, and information security. In his new role, he will also help strengthen the enterprise architecture, network and information security. Mr. Carey entered the Senior Executive Service in June 2003 as the DON Deputy Chief Information Officer (Policy and Integration) and was responsible for leading the DON CIO staff in developing strategies for achieving IM/IT enterprise integration across the Department. Mr. Carey's Federal service began with the U.S. Army at the Aberdeen Proving Ground in October 1982 where he worked as a Test Director evaluating small arms and automatic weapons and their ammunition. He began his service with the Department of the Navy in February 1985 with the Naval Sea Systems Command. He worked in the Anti-Submarine/Undersea Warfare domain where he served in a variety of engineering and program management leadership positions within the Acquisition Community, culminating in his assignment as the Deputy Program Manager for the Undersea Weapons Program Office. Mr. Carey joined the staff of the DON CIO in February 2000, serving as the DON CIO eBusiness Team Leader through June 2003. During this period he also served as the Director of the DON Smart Card Office from February through September 2001. Mr. Carey attended the University of South Carolina where he received a Bachelor of Science degree in Engineering in 1982. He earned a Master of Engineering Management degree from the George Washington University in 1995. He has been a member of the Acquisition Professional Community and has been awarded the Department of the Navy Distinguished Civilian Service Award (twice) as well as the Superior and Meritorious Civilian Service Awards, and numerous other performance awards. He received the prestigious Federal 100 Award in 2006, 2008 and 2009 recognizing his significant contributions to Federal information technology. Mr. Carey was also named Department of Defense Executive of the Year for 2009 by Government Computer News. Mr. Carey is an active member of the United States Navy Reserve and currently holds the rank of CAPTAIN in the Civil Engineer Corps. He was recalled to active duty for Operation Desert Shield/Storm and Operation Iraqi Freedom, where, in 2006-2007, he served in the Al Anbar province with I Marine Expeditionary Force.

- Maj Gen Ronnie D. Hawkins, Jr., USAF, *Vice Director, Defense Information Systems Agency*

Maj Gen Ronnie D. Hawkins, Jr., is the Vice Director of the Defense Information Systems Agency (DISA). As Vice Director, he helps lead a worldwide organization of more than 6,600 military and civilian personnel responsible for planning, developing, and providing interoperable, global net-centric solutions that serve the needs of the president, secretary of defense, Joint Chiefs of Staff, the combatant commanders, and other Department of Defense (DoD) components. Maj Gen Hawkins received his commission as a distinguished graduate of the ROTC program at Angelo State University in 1977. He has held a variety of communications positions, including an assignment on the Joint Staff as support manager for command, control, communications and computer systems, and he later served as Director of C4 Systems for Joint Task Force - Southwest Asia. The general has commanded Cadet Squadron 24 at the U.S. Air Force Academy; Air Combat Command's Computer Systems Squadron and Communications Group; and Air Force Officer Accession and Training Schools at Maxwell Air Force Base, Ala. He has served as the Director of Communications and Information, Headquarters Pacific Air Forces, and Director of Communications Operations, Office of the Deputy Chief of Staff for Installations and Logistics, Headquarters U.S. Air Force. Maj Gen Hawkins has also been Deputy Chief of Staff, Communications and Information Systems, Multi-National Force-Iraq.

- Mr. Ron Bechtold, *Chief Information Officer, OSD*

Mr. Bechtold provides Information Technology operational and technical support to the Office of the Secretary of Defense, Chief Information Officer in designing, implementing, and maintaining information technology solutions for the Office of the Secretary of Defense. His responsibilities include, but are not limited to: Advising the Director, Washington Headquarters Services (WHS) on all on IT/IM operational matters pertaining to OSD; operating OSD's enterprise IT infrastructure services including, but not limited to, email, Remote Access Services (RAS) / Virtual Private Networks (VPNs), wireless/Blackberry, desktop computers, servers, storage systems, backup systems, and helpdesks; performing engineering services for OSD's enterprise IT infrastructure services including, but not limited to, email, RAS/VPN, wireless/Blackberry, desktop computers, servers, storage systems, backup systems, and helpdesks; operating and maintaining OSD's IT/IM Continuity of Operations (COOP), Continuity of Government (COG), and Continuity of Business (COB) systems; implement an OSD IT OSD Information Assurance (IA) program and infrastructure in accordance with the Defense Information Technology Security Certification and Accreditation Program (DITSCAP) / Defense Information Assurance Certification and Accreditation Program (DIACAP) that includes an IA planning process, Certification & Accreditation (C&A), IA awareness and training, and appropriate resource management; performing IT/IM project execution and operations in conformance

FEATURED SPEAKER PROFILES - *Listed in Order of Appearance*

with OSD CIO plans, including the enterprise architecture, strategic plans, annual plans, and metrics plans; executing acquisition and contract actions to support the OSD IT/IM program in accordance with OSD CIO direction; executing the OSD IT/IM budget in accordance with OSD CIO direction; performing lifecycle IT/IM asset management including purchasing, inventorying, and disposal of IT assets; designing new enterprise IT architectures or infrastructures. Includes the design, building and implementation of major new IT and communications services for all OSD components to include SecDef Communications.

- Mr. Daniel F. McMillin, *Deputy Chief, Warfighting Integration; Deputy Chief Information Officer, Office of the Secretary of the Air Force*

Daniel F. McMillin, a member of the Senior Executive Service, is Deputy Chief, Warfighting Integration, and Deputy Chief Information Officer, Office of the Secretary of the Air Force, Washington, D.C. Mr. McMillin entered federal civil service in 1983 as an auditor with the Defense Contract Audit Agency. From 1984 to 1986, he served as an operations and maintenance budget analyst for Naval Air Systems Command, and then as procurement budget analyst for the Office of the Comptroller of the Navy. For the next five years, Mr. McMillin performed duties as Management Support Director and Executive Director at the Naval Plant Representative Office in Melbourne, Australia. He returned to the Pentagon as a budget analyst for the Defense Business Operations Fund in the Office of the Comptroller of the Navy until 1992. From 1992 to 1997, Mr. McMillin served as Technical Director, then as Deputy Director, for Program Analysis and Financial Management at U.S. Transportation Command. He was appointed to the Senior Executive Service in August 1997 as USTRANSCOM's Deputy Director for Plans and Policy. He has been assigned to Headquarters U.S. Air Force as Associate Director of Programs, and previously served in the Office of Warfighting Integration and Chief Information Officer as Director, Policy, Planning and Resources. Prior to his current assignment, he was Deputy Director of the Air Force Staff.

- Mr. Gary L. Winkler, *United States Army Program Executive Officer, Enterprise Information Systems*

Mr. Gary Winkler took command of the Program Executive Office for Enterprise Information Systems (PEO EIS) in October 2007. In this assignment, he is responsible for large-scale Department of Defense (DoD) and Army Information Technology (IT) system development efforts supporting finance, logistics, personnel, communications infrastructure, biometrics, medical and war-fighting functions. He leads a workforce of more than 2,600 military, civilian and contractor personnel around the world, effectively executing approximately $4 billion or about 56 percent of the Army's FY10 IT budget. Mr. Winkler began his DoD career as a college student and Engineering Technician for the Army's Night Vision Lab. After graduate school, he went to work in private industry for the LTV Aerospace & Defense Company in Dallas as a Senior Investment Analyst responsible for Capital Planning/Budgeting, Investment Analysis, and Program Economics. He later moved back to Virginia where he worked for smaller companies providing technical services to DoD programs. He returned to the Army in the PEO for Command and Control Systems where he worked in various capacities on intelligence systems, culminating as Software Division Chief and Software Product Manager for the All Source Analysis System. He had two more follow-on PM assignments for ACAT IAM programs, and had assignments as an Acquisition Specialist at HQDA and Deputy PEO in the USAF PEO for Joint Logistics. Mr. Winkler was appointed to the Senior Executive Service in 2003, working in Army Headquarters under the Chief Information Officer/G6, as the Army's first Chief Knowledge Officer (CKO) and Director for Governance and Acquisition. In this capacity, he was responsible for Information Technology and Knowledge Management policies, programs and systems. Additionally, he led the Army's IT Human Capital Development program. Mr. Winkler holds Electrical Engineering and Mathematics degrees from Virginia Tech, an MBA from William and Mary, and a Master's Degree in National Resource Strategy from the Industrial College of the Armed Forces. Mr. Winkler's federal service awards include Presidential Rank Awards (Distinguished Executive Rank 2007, Meritorious Executive Rank 2009); the Secretary of the Army's Decoration for Exceptional Civilian Service (2006), the Army's Meritorious Civilian Service Award (2003), and the Army's Superior Civilian Service Award (2000, 1996).

MONDAY AT A GLANCE

"What's All This Agile Stuff About, Anyway?"

7:00 am - 4:00 pm

REGISTRATION OPEN - Maryland Suite Foyer

7:30 am - 8:30 am

CONTINENTAL BREAKFAST - Maryland Suite Foyer

8:30 am - 4:00 pm

CONCURRENT TRACKS - See Track Layout

10:00 am - 10:15 am

BREAK - Maryland Suite Foyer

11:30 am - 1:00 pm

LUNCH - Harborview

2:30 pm - 2:45 pm

BREAK - Maryland Suite Foyer

4:00 pm

ADJOURNED FOR THE DAY

TUESDAY AT A GLANCE

"The Means for DoD to change the Acquisition of Information Systems"

7:00 am - 6:15 pm

REGISTRATION OPEN - Maryland Suite Foyer

7:00 am - 8:00 am

CONTINENTAL BREAKFAST - Maryland Suite Foyer

8:00 am - 11:30 am

GENERAL SESSION - Maryland Suite: Columbia/Frederick

9:45 am - 10:00 am

BREAK - Maryland Suite Foyer

11:30 am - 1:00 pm

NETWORKING LUNCH - Harborview

1:00 pm - 5:15 pm

CONCURRENT TRACKS - See Track Layout

3:00 pm - 3:15 pm

BREAK - Maryland Suite Foyer

5:15 pm - 6:15 pm

NO HOST WELCOME RECEPTION - Harborview

6:15 pm

ADJOURNED FOR THE DAY

MONDAY, APRIL 4

8:30 am - 11:30 am

TRACK A Maryland Suite: Annapolis	**THE AGILE PROCESS:** *Iterative incremental lifecycle; lightweight processes including scrum and tools* Mr. Jeff Payne, *CEO and Founder, Coveros, Inc.*
TRACK B Maryland Suite: Baltimore	**AGILE REQUIREMENTS:** *User stories, non-functional requirements, the product backlog* Dr. Ahmed Sidky, *Executive Vice President, Santeon Group*

1:00 pm - 4:00 pm

TRACK A Maryland Suite: Annapolis	**AGILE DEVELOPMENT:** *Test first development, pair programming, automated unit testing, continuous integration, refactoring* Mr. Jeff Payne, *CEO and Founder, Coveros, Inc.*
TRACK B Maryland Suite: Baltimore	**AGILE TESTING:** *Agile testing quadrants, automation, exploratory testing, requirements expressed as tests* Mr. Nate Oster, *Agile Player-Coach and Founder, CodeSquads LLC*

TUESDAY, APRIL 5

8:00 am **WELCOME REMARKS**

- Dr. Steve Kimmel, *Chairman, NDIA C4ISR Division; Senior Vice President, Alion Science & Technology*

8:15 am **FEATURED GOVERNMENT PRESENTATION: DoD CYBER OPERATIONS - SECURING THE NATION IN THE DIGITAL ERA**

- Mr. Don Boian, *Technical Director, Operations, USCYBERCOM*

9:00 am **INDUSTRY KEYNOTE ADDRESS: THE PROMISE OF AGILE DEVELOPMENT**

- Mr. Sanjiv Augustine, *President, LitheSpeed*

10:00 am **THE DEMAND: INFORMATION SHARING — THE USER PERSPECTIVE OF WARFIGHTER NEEDS AND EXPECTATIONS — OPERATIONALIZING THE CLOUD**

- COL Timothy P. Hill, USA, *Director, Futures, Army Intelligence and Security Command (INSCOM)*

10:45 am **EARNED VALUE MANAGEMENT + AGILE DEVELOPMENT = IT PROGRAM SUCCESS**

- Mr. Glen Alleman, *Principle of Practices, Lewis & Fowler*

1:00 pm - 2:00 pm

TRACK A Maryland Suite: Annapolis	**SCRUM AND THE SCRUM MASTER** Mr. Sanjiv Augustine, *President, LitheSpeed*
TRACK B Maryland Suite: Baltimore	**AGILE PLANNING AND ESTIMATION** Dr. Ahmed Sidky, *Executive Vice President, Santeon Group*

TUESDAY, APRIL 5

2:00 pm - 3:00 pm

TRACK A Maryland Suite: Annapolis	**THE ROLE OF THE PRODUCT OWNER** Mr. Mike Cox, *Senior Consultant, Net Objectives*
TRACK B Maryland Suite: Baltimore	**MANAGING THE AGILE TEAM** Dr. Ahmed Sidky, *Executive Vice President, Santeon Group*

3:15 pm - 4:15 pm

TRACK A Maryland Suite: Annapolis	**CREATING AN EFFECTIVE BACKLOG** Dr. Ahmed Sidky, *Executive Vice President, Santeon Group*
TRACK B Maryland Suite: Baltimore	**RISK MANAGEMENT IN AGILE** Mr. Sanjiv Augustine, *President, LitheSpeed*

4:15 pm - 5:15 pm

TRACK A Maryland Suite: Annapolis	**SECURE AGILE DEVELOPMENT** Mr. Jeff Payne, *CEO and Founder, Coveros, Inc.*
TRACK B Maryland Suite: Baltimore	**LEAN AND KANBAN** Mr. Mike Cox, *Senior Consultant, Net Objectives*

WEDNESDAY, APRIL 6

8:00 am **WELCOME REMARKS**

- Dr. Steve Kimmel, *Chairman, NDIA C4ISR Division; Senior Vice President, Alion Science & Technology*

8:15 am **GOVERNMENT KEYNOTE ADDRESS**

- Honorable Frank Kendall, *Principal Deputy Under Secretary of Defense, AT&L, OSD*

9:00 am **IT ACQUISITION**

- Mr. David M. Wennergren, *Assistant Deputy Chief Management Officer, Department of Defense*

10:00 am **DoD CIO PANEL: DoD IMPLEMENTATION OF AGILE DEVELOPMENT**

Panel Chair: Mr. Rob Carey, *Deputy Chief Information Officer, Department of Defense*

Panelists:

- Maj Gen Ronnie Hawkins, Jr., USAF, *Vice Director, DISA*
- Mr. Ron Bechtold, *Chief Information Officer, OSD*
- Mr. Dan McMillin, *Deputy Chief Information Officer, Air Force*
- Mr. Gary Winkler, *Army PEO, EIS*

11:45 am **CLOSING REMARKS**

WEDNESDAY AT A GLANCE

"DoD Information Systems Acquisition"

7:00 am - 12:00 pm

REGISTRATION OPEN - Maryland Suite Foyer

7:00 am - 8:00 am

CONTINENTAL BREAKFAST - Maryland Suite Foyer

8:00 am - 12:00 pm

GENERAL SESSION - Maryland Suite: Columbia/Frederick

9:45 am - 10:00 am

BREAK - Maryland Suite Foyer

12:00 pm

CONFERENCE ADJOURNED

THANK YOU TO OUR SPONSOR

FedSources
A WASHINGTON MANAGEMENT GROUP COMPANY

FedSources Market Intelligence Services combines human analysis with government intelligence to drive growth within your company. Our strength is in our people. As a client, you'll get unlimited access to our team of government experts who will answer any question, any time.

FedSources provides a depth and breadth of government intelligence unmatched in the industry. We do the research and analysis for you. We deliver detailed intelligence - from agency spending priorities to targeted opportunities to specific government contacts - along with the "human touch" necessary to align that intelligence with your company's objectives.

More than 80% of our staff is dedicated to research and client support. Our government experts cover a broad range of professional services industries, from information technology to energy to healthcare. No matter what your industry focus, we provide the intelligence you need to target the right customers and identify ideal opportunities.

At FedSources, your business growth goals are our measures of success. Find out more at www.fedsources.com.

THANK YOU TO OUR SUMMIT PARTNER

SOFTWARE QUALITY ENGINEERING

Software Quality Engineering delivers training, support, research and publications to software managers, developers, test professionals, and quality engineers worldwide.

Since 1986, Software Quality Engineering has been at the forefront of software quality improvement technology, and was instrumental in setting the stage for the software industry to view testing as a distinct discipline.

Today, Software Quality Engineering produces several of the most respected conferences in the software testing industry and provides testing and development training for more than half of the Fortune 1000. They also produce some of the industry's highest-rated publications — *Better Software* magazine and StickyMinds.com.

For more information about Software Quality Engineering, visit them on the web at www.sqe.com.

Agile Development

Jeffery Payne
CEO, Coveros, Inc.
jeff.payne@coveros.com

Agenda

- Introductions & Expectations
- What is Agile?
- Agile Development Planning
- Agile Development Iterations
- Wrap Up

About Coveros

- Coveros helps organizations accelerate the delivery of secure, reliable software

- Our consulting services:
  - Agile software development
  - Application security
  - Software quality assurance
  - Software process improvement

- Our key markets:
  - Financial services
  - Healthcare
  - Defense
  - National security

Corporate Partners

ARXAN

FORTIFY SOFTWARE

Telos

SQE TRAINING

Introductions

Instructor – Jeffery Payne

Jeffery Payne is CEO and founder of Coveros, Inc., a software company that helps organizations accelerate the delivery of secure, reliable software. Coveros uses agile development methods and a proven software assurance framework to build security and quality into software from the ground up. Prior to founding Coveros, Jeffery was Chairman of the Board, CEO, and co-founder of Cigital, Inc. Under his direction, Cigital became a leader in software security and software quality solutions, helping clients mitigate the risk of software failure. Jeffery is a recognized software expert and popular speaker at both business and technology conferences on a variety of software quality, security, and agile development topics. He has also testified before Congress on issues of national importance, including intellectual property rights, cyber-terrorism, Software research funding, and software quality.

Class Attendees

Expectations

- What are your expectations for this class?
- What do you wish to learn?
- What questions do you want answered?

Objectives

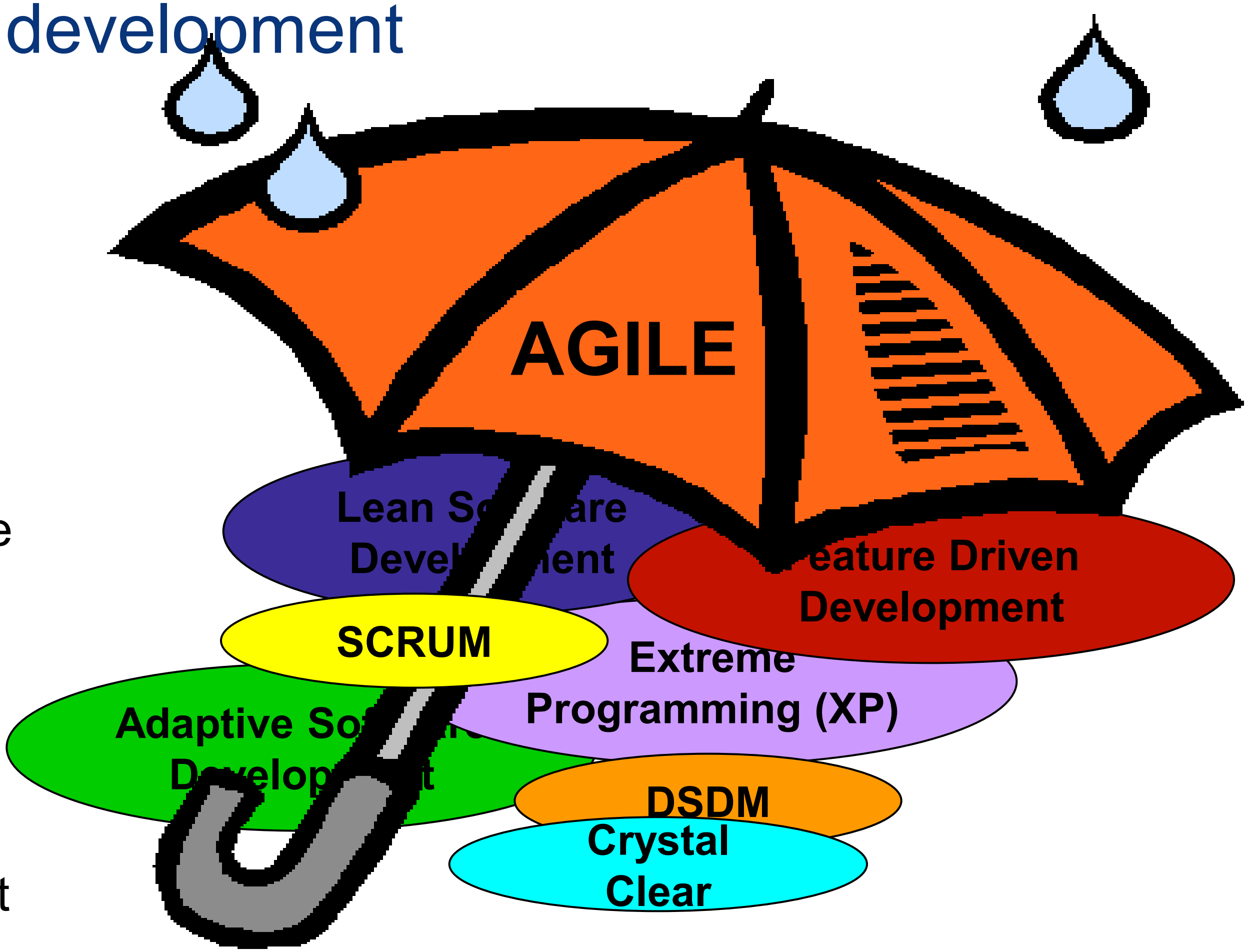
The primary objectives of this course are to:

- Introduce you to Agile software development
- Outline the major steps required to successfully plan and execute an Agile software project.
- Provide an overview of the leading Agile development best practices

What is Agile?

The agile movement began as a set of ideas for improving software development

- Close collaboration between programmers & business people
- Face-to-face communication
- Frequent delivery of deployable business value
- Self-organizing teams
- Crafting code & environment to support requirements changes
- The most important output of a project is working software


AGILE
Lean Software Development
Feature Driven Development
SCRUM
Extreme Programming (XP)
Adaptive Software Development
DSDM
Crystal Clear

http://www.agilemanifesto.org

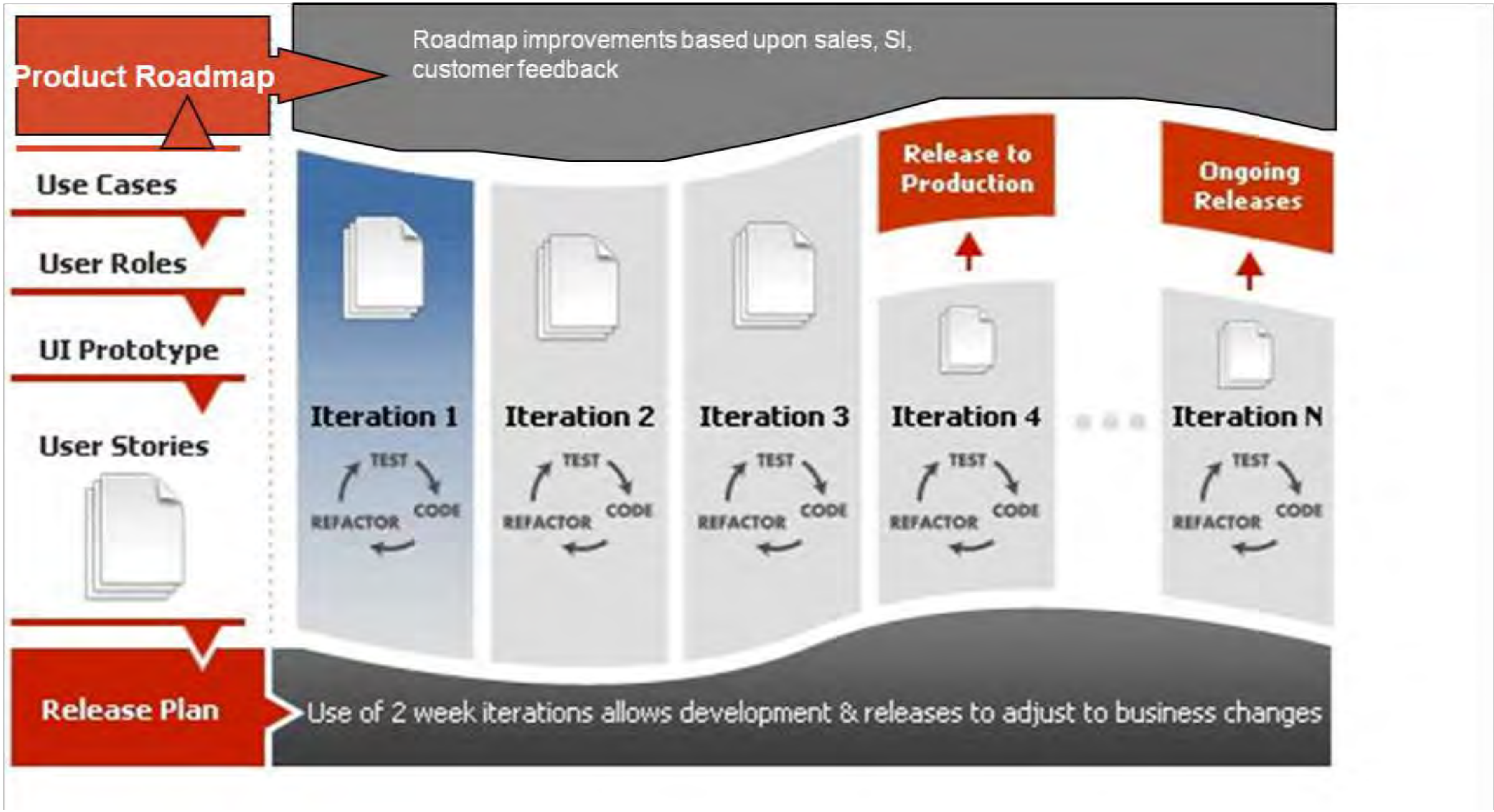
What is Agile?

All agile methodologies adhere to some basic principles

- Early and continuous delivery of valuable software
- Welcome changing requirements, even late in development
- Deliver working software frequently
- Business people and developers work together daily
- Build projects around motivated individuals and trust them to get the job done.
- Frequent conversation to convey information efficiently
- Working software as the primary measure of progress
- Sustainable development
- Continuous attention to technical excellence and good design
- Simplicity—maximizing the amount of work not done
- The best architectures, requirements, and designs emerge from self-organizing teams
- At regular intervals, the team reflects on, tunes, and adjusts its behavior

Agile Product Development Process

Initial Planning On-going Planning, Implementation & Release



- Incremental product delivery process that encompasses all aspects of the organization
- Team-oriented with day-to-day interactions between all functions

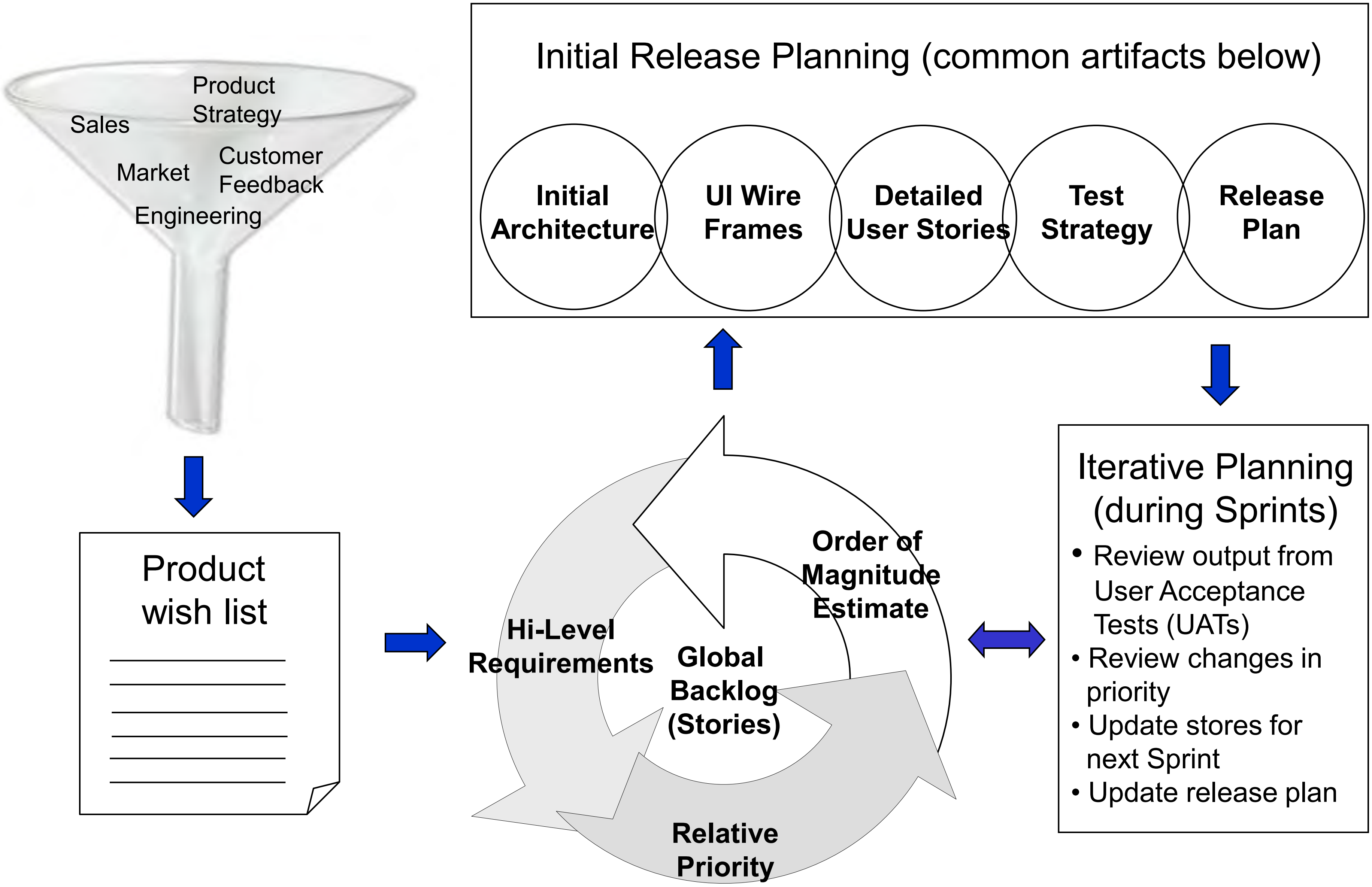
Agile Development Planning

Agile Development Planning

EXTREME PROGRAMMING
I CAN'T GIVE YOU ALL OF THESE FEATURES IN THE FIRST VERSION.
AND EACH FEATURE NEEDS TO HAVE WHAT WE CALL A "USER STORY."
OKAY, HERE'S A STORY: YOU GIVE ME ALL OF MY FEATURES OR I'LL RUIN YOUR LIFE.
www.dilbert.com scottadams@aol.com
1/6/03 © 2002 United Feature Syndicate, Inc.

Copyright © 2003 United Feature Syndicate, Inc.

Agile Planning Process



Product Strategy

Sales

Customer Feedback

Market

Engineering

Product wish list

Initial Release Planning (common artifacts below)

Initial Architecture

UI Wire Frames

Detailed User Stories

Test Strategy

Release Plan

Hi-Level Requirements

Order of Magnitude Estimate

Global Backlog (Stories)

Relative Priority

Iterative Planning (during Sprints)

- Review output from User Acceptance Tests (UATs)
- Review changes in priority
- Update stores for next Sprint
- Update release plan

Creating a Global Backlog

- A Global Backlog encompasses all items from the Wish List that Product Management deems the highest priority for inclusion in upcoming releases.
- Backlog items are prioritized and assigned an Order of Magnitude (OOM) estimate
- OOM can be used for budgeting purposes BUT ONLY IF THE TOP END VARIANCE ESTIMATE IS USED
- A Global Backlog contains User Stories

User Stories

- A story represents some small slice of visible, usable functionality—typically something a user can do with the system
- A well-written story possesses the following characteristics:
  - Understandable
  - Testable
  - Valuable to the customer
  - Independent of each other
  - Small enough to build a handful each Sprint
- Stories are written during initial planning or during a Sprint planning meeting once the project has begun.
- Although the idea for a story will most likely originate from the business stakeholders, many team members may have a hand in authoring the story card, including project managers, tech leads, analysts, and testers.

Initial Release Planning

- Planning within Agile is iterative
- Regardless, some planning must be done up front for a variety of reasons:
  - Build a release plan the organization can plan around
  - Resolve upfront architectural tradeoffs so implementation conforms to an overall architectural vision
  - Prototype / wireframe a UI to get early feedback from stakeholders on the requirements
  - Prepare development & testing platforms accordingly
- Detail out initial Sprint(s) and projected releases for more detailed budgeting purposes

Roles of Core Planning Team Members

- *Business Stakeholder(s)* – Own the product vision and helps make sure the requirements meet the needs of the end customer
- *Project Manager* – Responsible for the end-to-end planning process
- *Lead Architect* – Responsible for the initial architecture and scoping
- *Lead Business Analyst* – Acts as SME and documents requirements
- *QA Lead* – Responsible for overall test strategy
- *Web Designer* – Optional depending upon whether UI prototyping is involved

Others participate in the process as required and necessary

Building a Release Plan

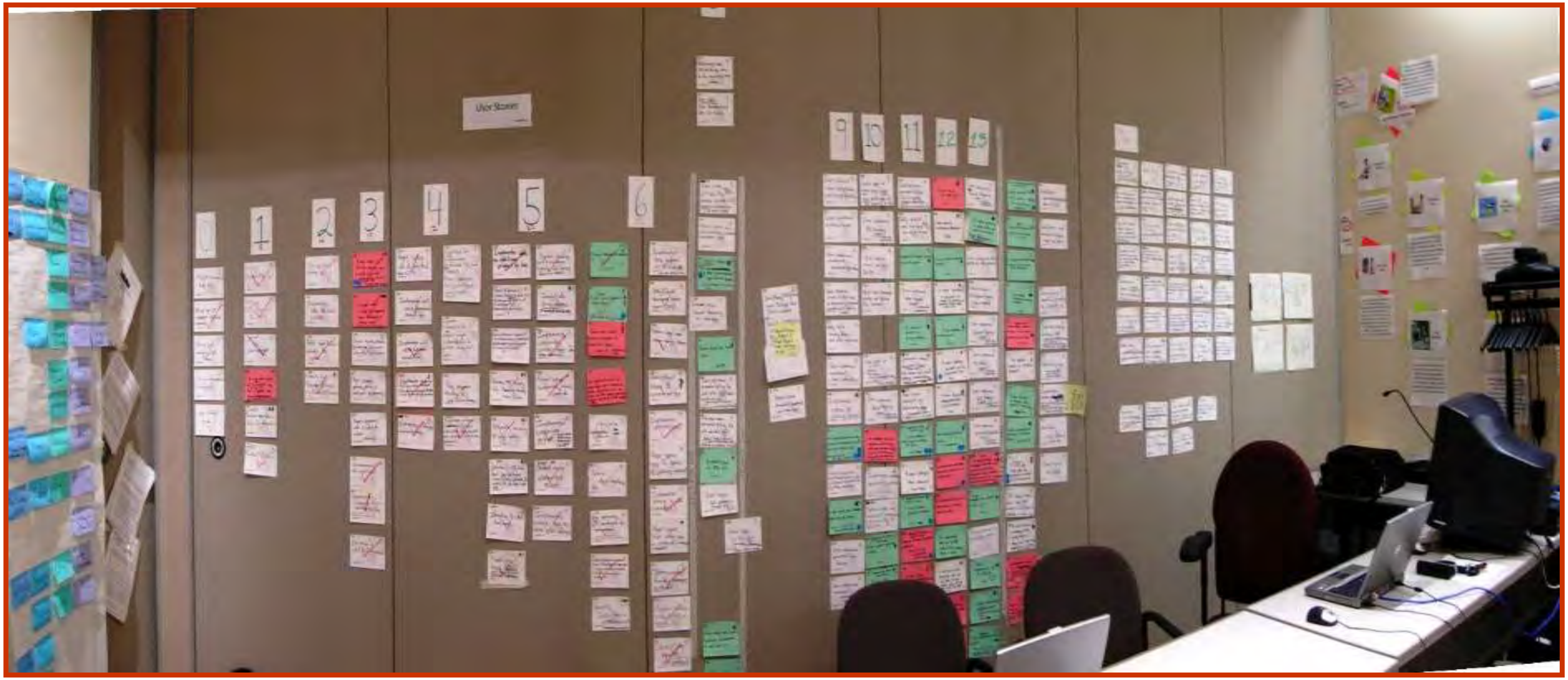
- Release plans that include Sprints* are built using:
  - Prioritized stories that include Points estimates
  - Team size
  - A decision around Sprint duration
  - A decision around how much functionality is enough to justify a release

- Sprint duration
  - Tradeoff between cost of change and organization's agility
  - Typically 2 – 4 weeks in duration

- Release decision
  - Tradeoff between cost of deployment/release and market dynamics
  - Releases vary from daily to 3 months (huge variation!)
  - Releases are now a business decision

**A Sprint is a development iteration in SCRUM terminology*

The Team Room Approach to Release Plan Mgmt



Pros – very visible and tangible, great for co-located teams, easy to modify
Cons – not under version control, harder for distributed teams to visualize, takes space

Agile Development

Goals of Development Sprints

- Construct a piece of software that fully meets the demands of the stakeholders.
- Ensure that the software is high quality and production ready.
- Have a code base that is well architected, commented and easily maintainable.
- Establish a sustainable development process that matches the skills and work habits of your development organization.
- Establish continuous integration infrastructure in time for production deployment.

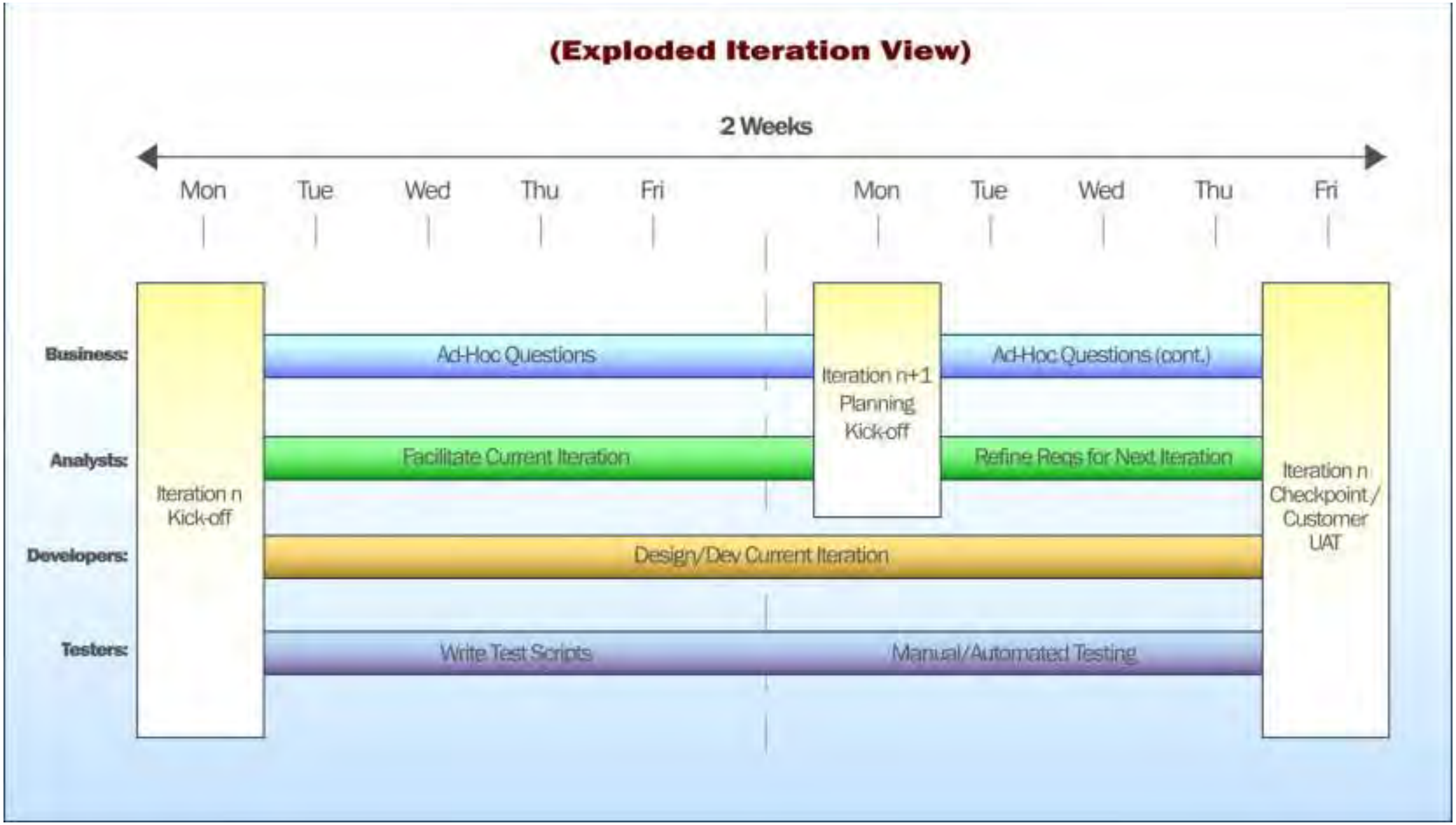
Am I Ready to Begin?

- Have Initial Planning deliverables been accepted by stakeholders?
- Are there at least one Sprints scheduled to full capacity?
- Is the management and development infrastructure established sufficiently to begin?
- Have the product stakeholders been identified and given full authority on the direction of the software to be developed?
- Is a <u>dedicated</u> development team identified?

Typical Roles

- Project Manager – responsible for the day to day functional delivery of the software, managing project schedule and priorities, and working with stakeholders to resolve any project issues
- Architect – responsible for coding, design and architecture standards review and compliance, solutions definition and overall performance characteristic of the software
- Analyst – supports project manager in the proper definition of requirements
- Development Lead – responsible for day to day technical implementation of the software and technical management of developers
- Developers – responsible for technical implementation of the software
- QA/Test Lead – responsible for the day to day testing, verification and validation of the software, compliance, management of the testers and automation of the test cases
- Testers – responsible for the testing, verification and validation of the software and the automation of the test cases
- Business stakeholder or proxy – available when needed to answer questions regarding the product, market, customer needs

Iterative Development Process


(Exploded Iteration View)
2 Weeks
Mon
Tue
Wed
Thu
Fri
Mon
Tue
Wed
Thu
Fri
Business:
Analysts:
Developers:
Testers:
Iteration n Kick-off
Ad-Hoc Questions
Facilitate Current Iteration
Iteration n+1 Planning Kick-off
Ad-Hoc Questions (cont.)
Refine Reqs for Next Iteration
Design/Dev Current Iteration
Write Test Scripts
Manual/Automated Testing
Iteration n Checkpoint / Customer UAT

Team Communication during Agile Development

- Effective communication between all team members is absolutely critical to a successful Agile project
- A meeting rhythm should be established to assure communication happens at least at key Sprint junctures
- Important team meetings include:
  - Sprint kickoffs
  - Daily standups
  - User acceptance testing
  - Retrospectives

Sprint Kickoff Meeting

- Occurs at the beginning of a new Sprint. May require as little as a couple of hours or a whole day.

- The purpose of the Sprint kickoff is to:
  - Allow the business team to communicate the very latest understanding of the scheduled stories.
  - Provide the development team with a chance to ask questions about the stories to the business team.
  - Allow the development team to break down the stories into tasks.
  - Enable the development team to refine the initial story estimates based on the tasks.

Daily Standup Meetings

- The daily standup occurs at the start of each day. Intended to last no more than 15 minutes.
- Participants are encouraged to actually ‗stand up‘ during the meeting so that the meeting stays short.¯
- The purpose is to allow each participant to quickly communicate:
  - What did I do yesterday?
  - What do I plan to do today?
  - What obstacles are standing in the way of achieving my goals today?
- One intent of this meeting is to identify potential issues as soon as possible.

Sprint Planning Meeting

- Sprint planning occurs sometime after the Sprint kickoff. In a two week Sprint, planning is ideally completed between the end of the first week and the beginning of the second week.
- The purpose of Sprint planning is to:
  - Analyze and discuss the stories that are scheduled for the next Sprint.
  - Adjust the list of scheduled stories based on various feedback channels, such as UAT.
  - Provide enough detail in the requirements so that the stories can be broken down into tasks during the next Sprint kickoff.

User Acceptance Testing (UAT)

- User Acceptance Testing usually occurs on the last day of the Sprint.
- This meeting:
  - Allows the stakeholders, in a hands on way, to use the features newly developed during the Sprint.
  - Allows the stakeholders to provide feedback on the features.
  - Identify bugs that may have been missed during development.
  - Typically spurs ideas for new features which go onto the Wish List

Communication during Agile Development

Retrospectives

- The retrospective takes place at the end of each Sprint, usually after the UAT.
- This meeting allows the team to talk about what went right and what went wrong during the Sprint.
- This meeting can often follow a fixed format (such as SAMOLO). But it‘s more important that it‘s conducted in a manner that encourages the participants to provide honest feedback.
- It is intended that the lessons learned in the retrospective are applied in future Sprints.
- Team members are held accountable for action items assigned during retrospective discussions.

SAMOLO

Common format for conducting a retrospective

- Same As (SA) – What should we keep doing the way we are doing it?
- More Of (MO) – What should we do more of than we‘ve done in the past?
- Less Of (LO) – What should we do less of than we‘ve done in the past?

- Other similar formats
  - Keep doing, Start doing, Stop doing
  - Thorns and Roses

The Role of Project Manager

- The Project Manager has final authority on all project decision

- During each Sprint, the PM:
  - Leads Sprint kickoff
  - Leads Daily Stand-ups
  - Leads UATs
  - Leads Retrospectives
  - Leads Iterative Planning Process
  - Removes Hurdles / Barriers from Team

- On small projects, a PM may also:
  - Participate in requirements definition
  - Participate in testing efforts

The Role of Business Analysts

- Business Analysts are responsible for assuring that requirements meet the needs of the customer

- During each Sprint, BAs:
  - Detail requirements for the next Sprint(s)
  - Provide subject matter expertise to development and testing on product requirements and customer needs
  - Review test plans for completeness

- On small projects a BA may also:
  - Participate in testing

The Role of Software Developers

- Software developers are responsible for software development
- During each Sprint, software developers:
  - Perform team-based design
  - Implement the application
  - Developer testing
  - Refactoring
  - Setup / maintain Continuous Integration environment
- On smaller projects a software developer may also:
  - Participate in software testing of functionality

Team-Based Design

- In team-based design, developers invest as little as possible in upfront design.
- They do not anticipate problems down the road that may or may not happen.
- Assume that the simplest design will work until proven otherwise.
- Involve all members of the team in system design. Multiple perspectives will ensure that as many potential issues are identified and addressed as early as possible.

Implementation – Coding Standards

- At the start of the project, the team should discuss which coding standards they agree to adopt. This can be a prickly issue.
- Some developers feel very strongly about how code should be written. Some goals of adopting coding standards:
  - Avoid petty disagreements during pair programming
  - Improve code readability
  - Enhance refactoring productivity (reducing the cost of change)
  - Enhance code maintainability
- Some of the most common coding standards address:
  - Naming conventions
  - Code organization and layout
  - Use of code comments
  - Avoidance of language specific problem constructs

Implementation – Pair Programming

- In pair programming, one developer works at the keyboard while the other follows along.
- The typing developer is focused on the code mechanics while the other is thinking at a broader level about what to do next. The second developer is also better able to spot bugs before they are deployed.
- This practice helps to spread domain and technical knowledge across the various members of the development team.
- Most teams prefer a balance between pair and individual programming that works best for them.

The Role of Software Developers

- The following benefits can be realized from effective pair programming:
  - Continuous code review
  - Cross pollination of developer knowledge
  - Increased code quality

Developer Testing – Test Driven Development

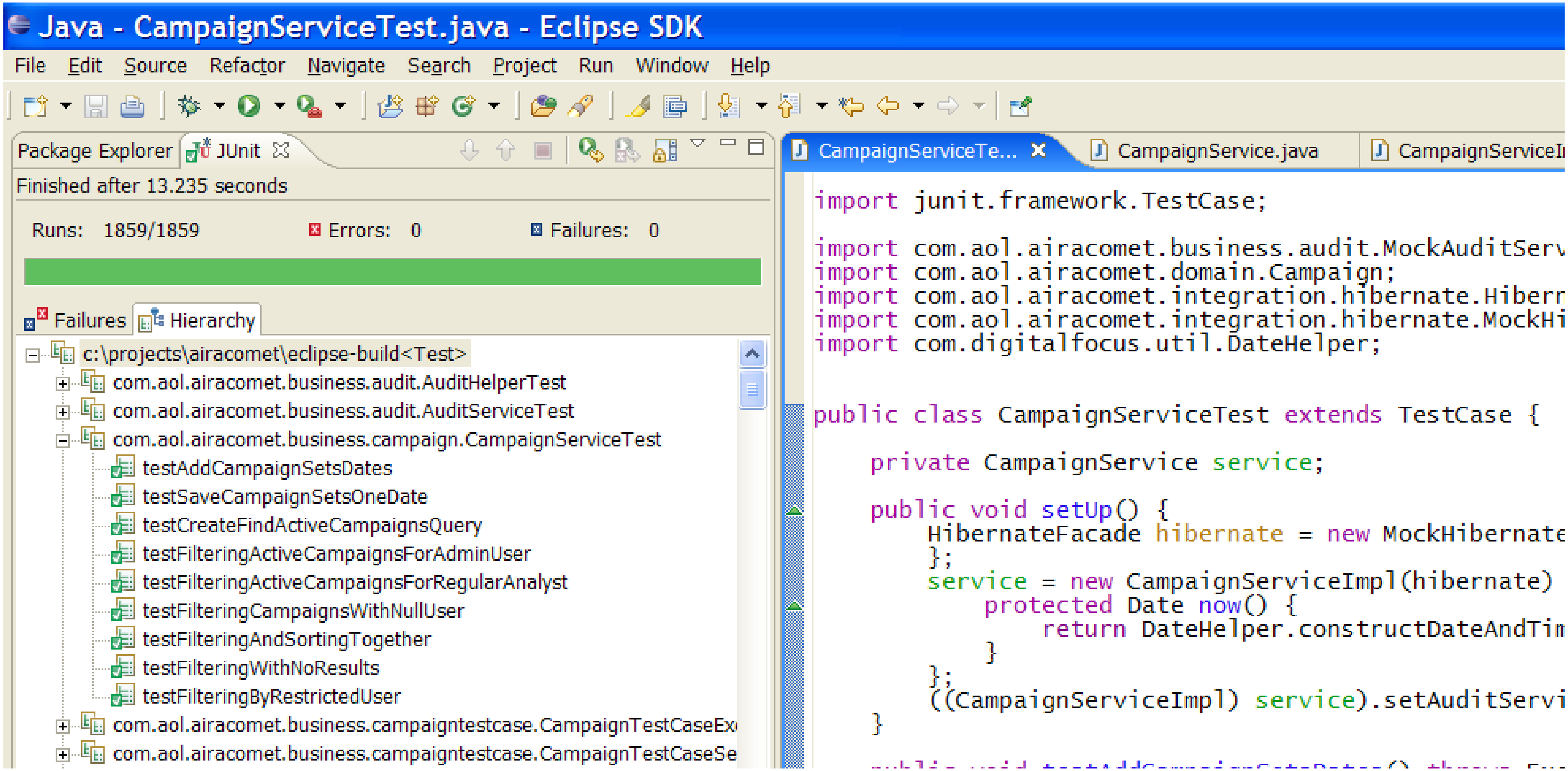
- Test Driven Development (TDD) is a software development technique in which developers consider tests as part of their specification when building software
  - Test First Development – creates tests before creating code
  - Test Influenced Development – outlines positive and negative test scenarios while thinking through implementation
- Tests are written to:
  - Test the functionality of units
  - Test interfaces between implemented components
  - Validate bug fixes
  - Validate refactoring
- Tests are automated for use during code build / test cycles

The Role of Software Developers

Developer Testing – Unit Testing

- A unit test is a piece of software that validates the correctness of a small unit of production code in a well-defined, repeatable manner.
- The adoption of unit tests in an agile environment hinges very much on the early availability of continuous integration tools.
- This allows the tests to be run continuously and gives the developers confidence to make changes to production code.
- Unit tests also serve as part of the documentation of the code.

Developer Testing – Automated Unit Testing


Java - CampaignServiceTest.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
CampaignServiceTe...
CampaignService.java
CampaignServiceI
Failures
Hierarchy

Finished after 13.235 seconds

Runs: 1859/1859 Errors: 0 Failures: 0

```
c:\projects\airacomet\eclipse-build<Test>
  com.aol.airacomet.business.audit.AuditHelperTest
  com.aol.airacomet.business.audit.AuditServiceTest
  com.aol.airacomet.business.campaign.CampaignServiceTest
    testAddCampaignSetsDates
    testSaveCampaignSetsOneDate
    testCreateFindActiveCampaignsQuery
    testFilteringActiveCampaignsForAdminUser
    testFilteringActiveCampaignsForRegularAnalyst
    testFilteringCampaignsWithNullUser
    testFilteringAndSortingTogether
    testFilteringWithNoResults
    testFilteringByRestrictedUser
  com.aol.airacomet.business.campaigntestcase.CampaignTestCaseEx
  com.aol.airacomet.business.campaigntestcase.CampaignTestCaseSe
```

```java
import junit.framework.TestCase;

import com.aol.airacomet.business.audit.MockAuditServ
import com.aol.airacomet.domain.Campaign;
import com.aol.airacomet.integration.hibernate.Hiberr
import com.aol.airacomet.integration.hibernate.MockHi
import com.digitalfocus.util.DateHelper;


public class CampaignServiceTest extends TestCase {

    private CampaignService service;

    public void setUp() {
        HibernateFacade hibernate = new MockHibernate
        };
        service = new CampaignServiceImpl(hibernate)
            protected Date now() {
                return DateHelper.constructDateAndTim
            }
        };
        ((CampaignServiceImpl) service).setAuditServi
    }

    public void testAddCampaignSetsDates() throws Exc
```

The Role of Software Developers

Refactoring

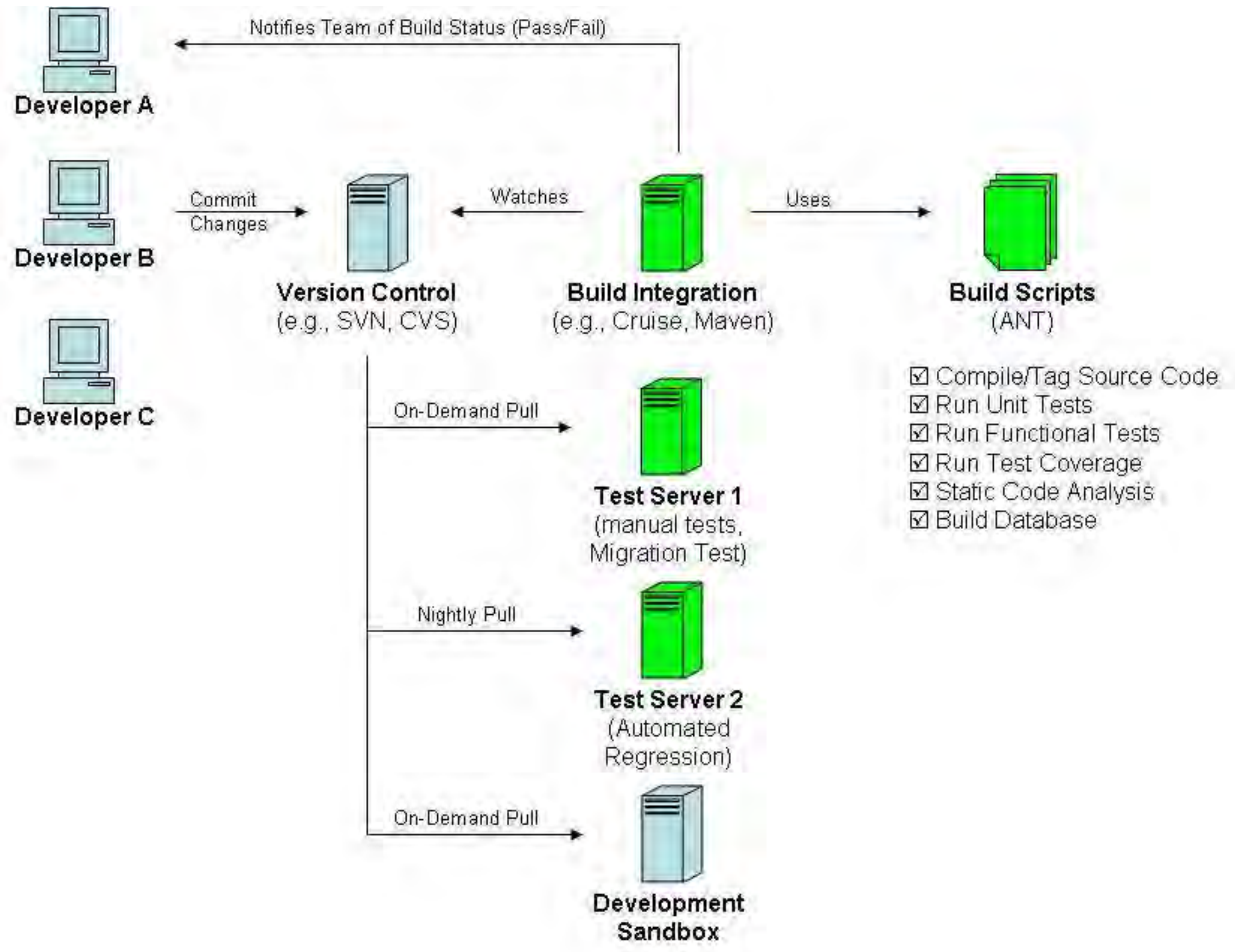
- As stories are completed, the code moves in directions that the development team did not anticipate.
- Sometimes code gets duplicated or unnecessarily complicated.
- If a developer is about to implement a story by increasing the amount of inefficient code, this is the appropriate time at which to refactor the code.
- Refactoring should be done in small amounts in order not to adversely impact the delivery schedule. Larger refactors should be broken up over one or more Sprints.

Continuous Integration

- Continuous Integration is the practice of frequently integrating the source code for a project or group of related or dependent projects
- The purpose of Continuous Integration is to keep code and build quality high and make delivery of the application or system easier because the build is performed enough to keep it clean and to work out any problems
- In a Continuous Integration process, after a successful build, a set of tests are run against the resulting software
- These tests range from unit and functional tests to integration, performance and security tests
- After the tests are run, many Continuous Integration processes apply code analysis tools to the code base in order to find code and security defects not detected by the tests.

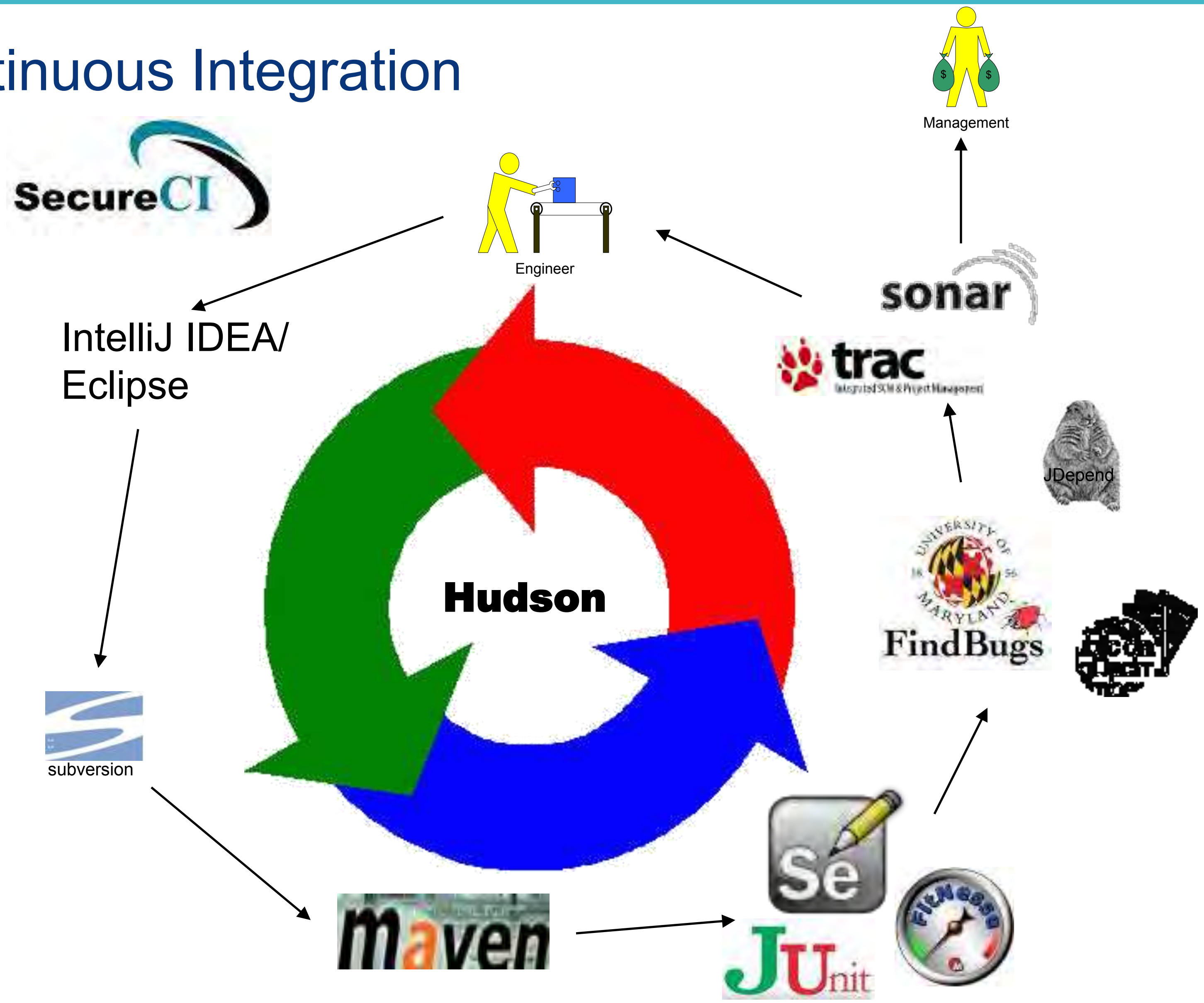
The Role of Software Developers

Continuous Integration



The Role of Software Developers

Continuous Integration



Management

Engineer

IntelliJ IDEA/ Eclipse

subversion

Hudson

JDepend

Automated Deployments

- The product team must carefully consider whether or not CI should be utilized to deploy to production.
- The team may choose not to use CI for production deployment for the following scenarios:
  - Large or complicated applications
  - Companies with policies that prevent auto deployment
  - Applications where the exact timing of the release must be strictly controlled
- The team might choose CI deployments to production under the following scenarios:
  - Applications with smaller user bases
  - Internal applications
  - Applications developed and owned by smaller companies

The Role of Software Testers

- Software testers are responsible for testing the application above and beyond developer testing
- During each Sprint, software testers:
  - Test software
  - Automate tests
  - Test plan for future Sprints
  - Analyze requirements for testability
- On smaller projects a software tester may also:
  - Participate in business analysis

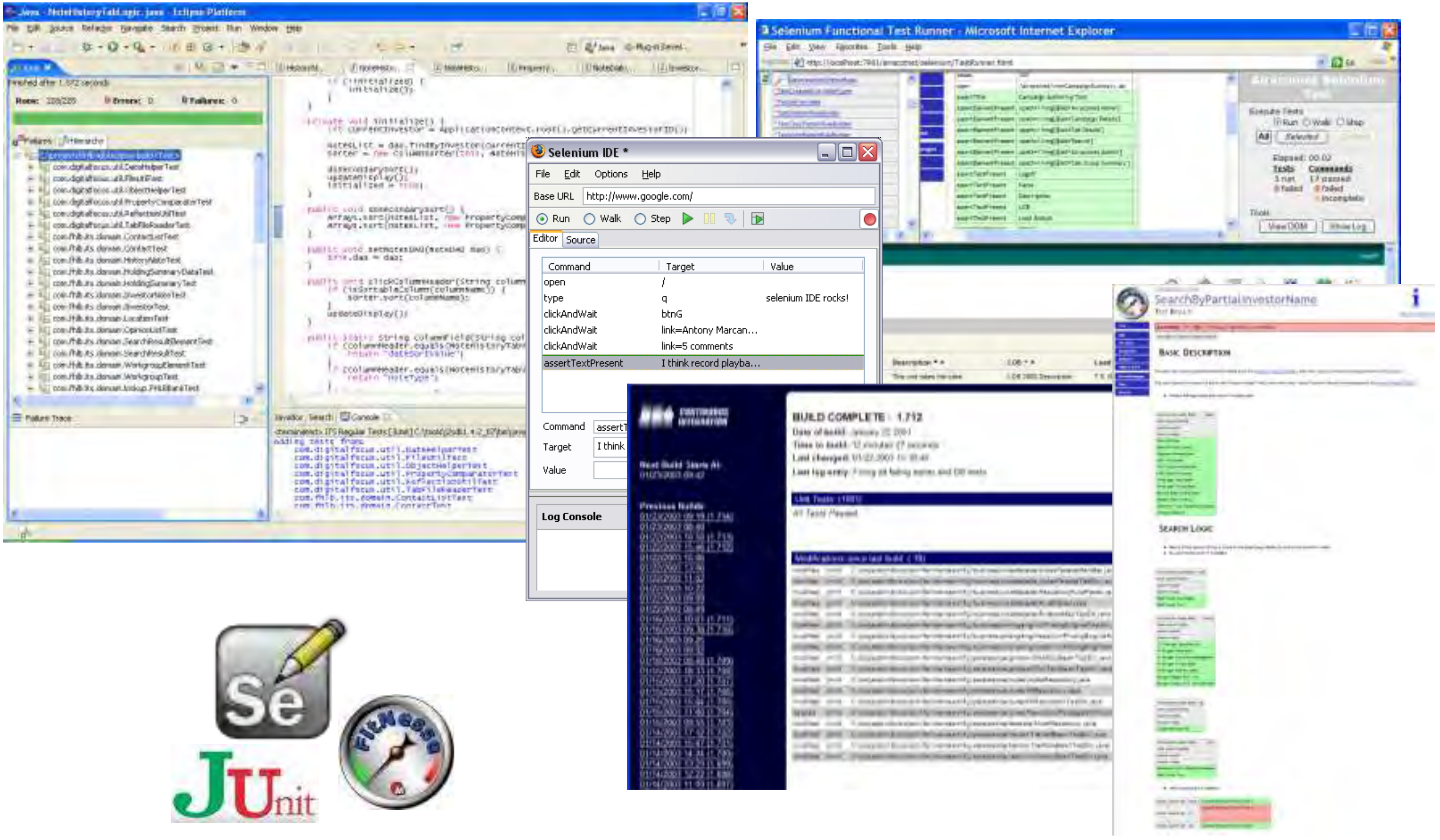
Software Testing – Agile Testing

- An essential part of Agile is continuous testing. Rather than delivering large amounts of untested code at the end of a Sprint or release, it is essential to test on a daily basis as the code is being developed.
- Software testing performed by software testers
  - Testing of key components, end-to-end stories, use cases, and feature sets for each Sprint
  - Testing of non-functional requirements (load/performance, security, fault-tolerance, etc.) for releases
  - Coordinate User Acceptance Testing and capture results
- Software testers work very closely with software developers on all testing tasks

Software Testing – Test Automation

- Effective test automation is achieved by:
  - Applying automation only where there is a clear ROI for doing so
  - Often times NOT test execution -> automating test setup, test results validation, test cleanup are often highly effective
  - A test must be run 3 – 10x unchanged before there is a return for automating it
- Structuring and treating test automation scripts as software that must be designed, developed, tested, and maintained
- Leveraging test automation infrastructure (both off-the-shelf and custom) across all appropriate development projects

Software Testing – Test Automation Tools



Base URL http://www.google.com/

Command	Target	Value
open	/	
type	q	selenium IDE rocks!
clickAndWait	btnG	
clickAndWait	link=Antony Marcan...	
clickAndWait	link=5 comments	
assertTextPresent	I think record playba...	

Command assertT

Target I think

Value

Software Testing – Which tests to automate?

- Part of the test planning process is deciding which tests or parts of tests to automate. Some criteria that should be considered:
  - Are the tests easy to automate? What makes a test easy to automate is the ability to script not only the behavior but also the analysis of the results to determine if the test passed or failed.
  - How often is the functionality or API point, used by the users or consumers of the product? - The more popular, prominent or useful the functionality under consideration is the more benefit in automating it.
  - How risky is the functionality? –No matter what the definition of risk, the goal in automating that functionality is to help mitigate the risk. One definition of risk is what features are hardest to implement correctly. The added assurance of automated tests can help mitigate the risk by having the tests run more frequently than they would without automation.
  - Is the cost of automating the functionality less than the cost of manual testing the functionality though the life of the project?

The Role of Business Customer / Proxy

- Represents the customer base on the project team
- During each Sprint, the business customer:
  - Answers ad-hoc questions on the product and its requirements
  - Participates in User Acceptance Testing
- Sometimes the appropriate business customer isn‘t available to be involved in the project. A business —proxy” acts on the business customer‘s behalf
  - Must have the authority to make decisions

Wrap-Up

Agile books we recommend

- Beck, Kent, —Extreme Programming – Embracing Change”, Addison-Wesley Professional, 2004
- Cohn, Rob, —User Stories Applied”, Addison-Wesley Professional, 2004
- Cohn, Rob, —Agile Estimating & Planning”, Prentice Hall PTR, 2005
- Crispin, Lisa, —Agile Testing – A Practical Guide for Testers & Agile Teams”, Addison-Wesley Professional, 2009
- Duvall, Paul, —Continuous Integration: Improving Software Quality and Reducing Risk”, Addison-Wesley Professional, 2007

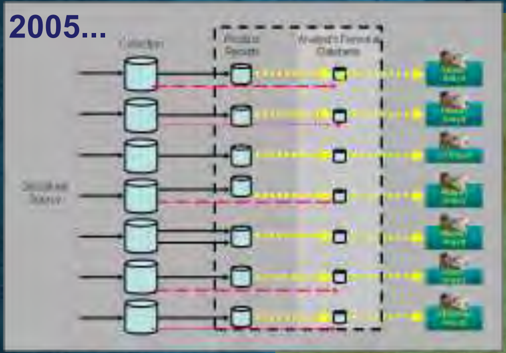
Questions?

- Contact information:
  - Jeffery Payne, Coveros Inc.
  - 703-431-2920
  - jeff.payne@coveros.com

CLOUD ANALYTICS:
Empowering the Army Intelligence Core Analytic Enterprise

5 APR 2011

Advanced Analytics
Harnessing Data for the Warfighter

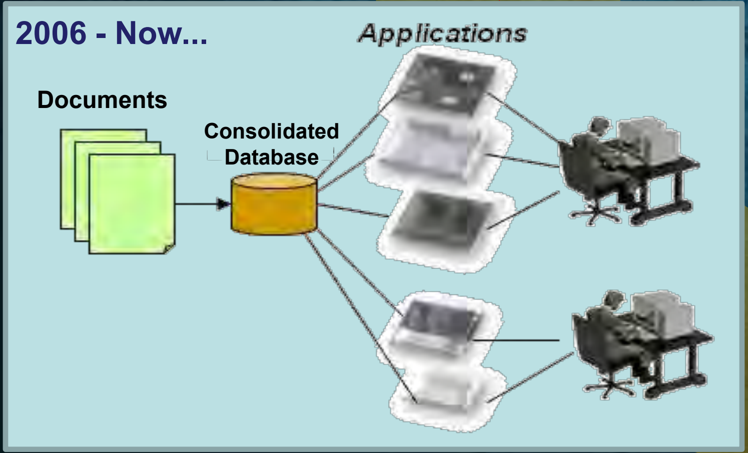


2005...

Data Silos

Global Enterprise “Common Architecture”

Data Access



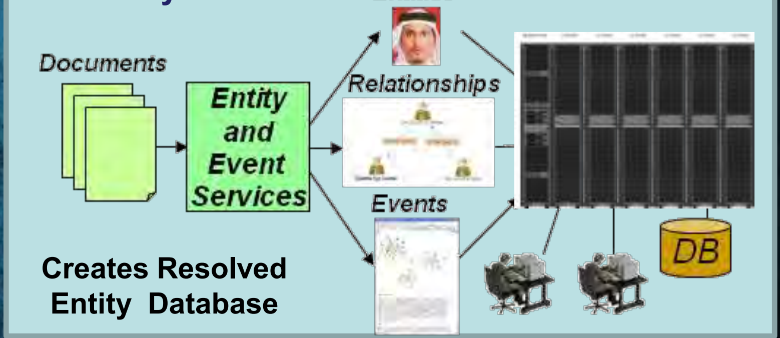
2006 - Now...

Document Centric

Data Deluge

2010 – Beyond...

Advanced Analytics



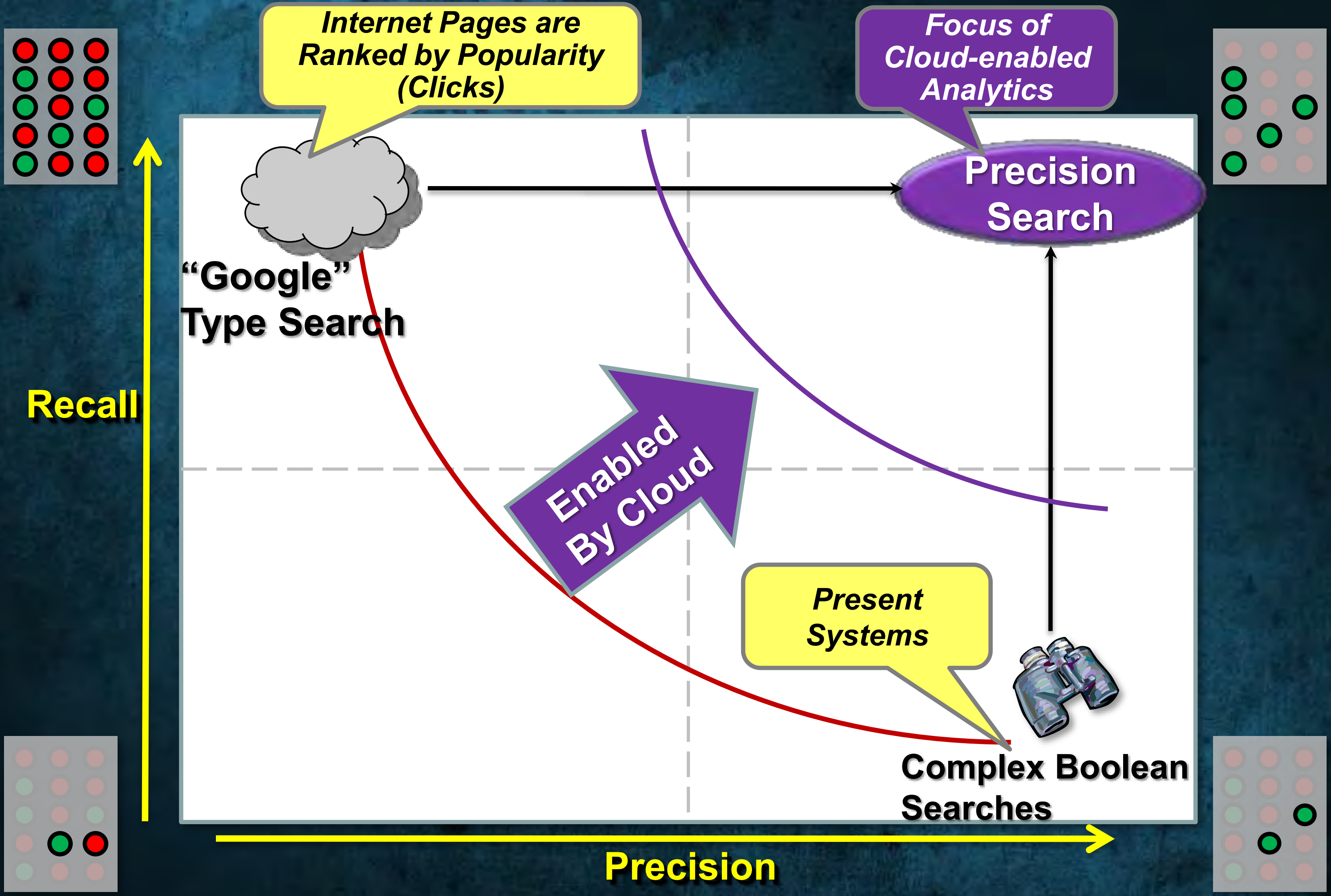
Creates Resolved Entity Database

Entity/Event Centric

- Focuses on Persons, Places, Activities, etc..)
- Resolves and Disambiguates Entities
- Extracts Entity-To-Entity Relationship
- Provides Entity & Event Representation

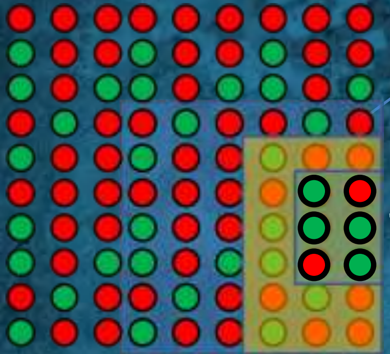
Precision Search

Finding Relevant Data Faster



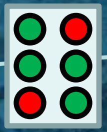
Working with ALL the Data

Today's Analytics Must Filter Data



All Data

Lexical Searches,
Boolean Search
Attributes, or Properties
Based Filters



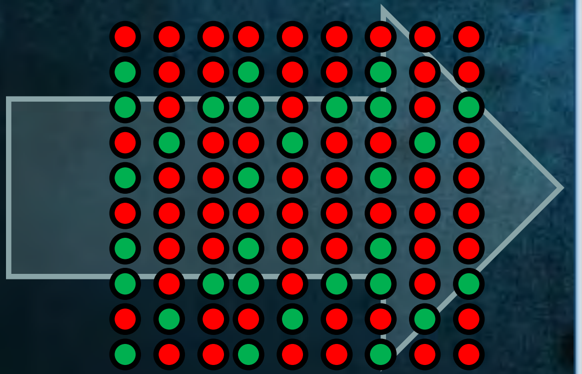
Working Set

Tools / Analysis

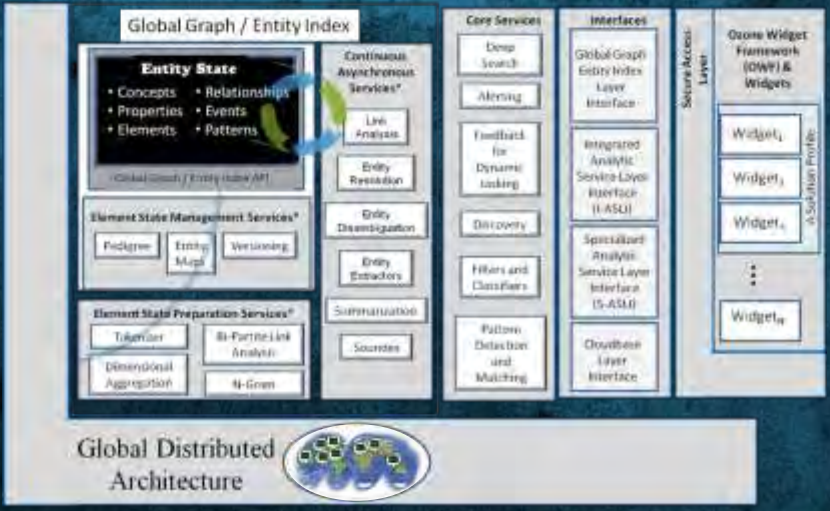
Product
from *__Filtered__* Data

With the Cloud, We Do Not Have to Filter to Get a Working Set

Cloud Analytics

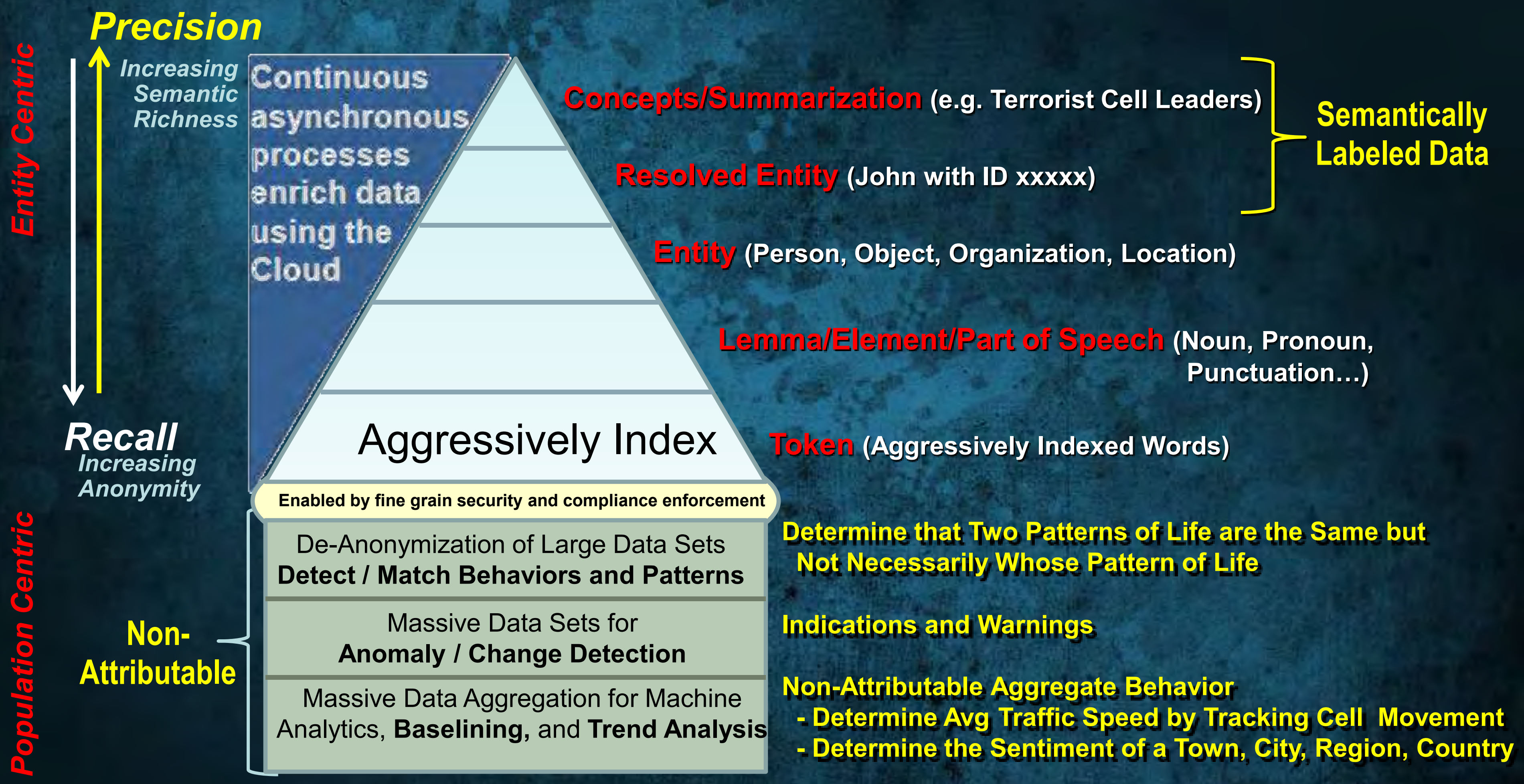


All Data Sources/Types


Global Graph / Entity Index
Entity State
• Concepts
• Relationships
• Properties
• Events
• Elements
• Patterns
Global Graph / Entity Index API
Element State Management Services*
Pedigree
Entity Maps
Versioning
Element State Preparation Services*
Tokenizer
Bi-Partite Link Analysis
Dimensional Aggregation
N-Gram
Continuous Asynchronous Services*
Link Analysis
Entity Resolution
Entity Disambiguation
Entity Extractors
Summarization
Sounders
Core Services
Deep Search
Alerting
Feedback for Dynamic tasking
Discovery
Filters and Classifiers
Pattern Detection and Matching
Interfaces
Global Graph Entity Index Layer Interface
Integrated Analytic Service Layer Interface (I-ASLI)
Specialized Analytic Service Layer Interface (S-ASLI)
Cloudbase Layer Interface
Secure Access Layer
Ozone Widget Framework (OWF) & Widgets
Widget$_1$
Widget$_2$
Widget$_3$
Widget$_N$
A Solution Profile
Global Distributed Architecture

Analysis Over the Entire Corpus – Follow the Analytic Threads *__without__* Re-Loading Data

Solving the Precision / Recall Conundrum with Semantic Enrichment of the Data



Entity Centric

Precision

Increasing Semantic Richness

Continuous asynchronous processes enrich data using the Cloud

Concepts/Summarization (e.g. Terrorist Cell Leaders)

Resolved Entity (John with ID xxxxx)

Semantically Labeled Data

Entity (Person, Object, Organization, Location)

Lemma/Element/Part of Speech (Noun, Pronoun, Punctuation…)

Aggressively Index

Token (Aggressively Indexed Words)

Recall

Increasing Anonymity

Enabled by fine grain security and compliance enforcement

De-Anonymization of Large Data Sets
Detect / Match Behaviors and Patterns

Determine that Two Patterns of Life are the Same but Not Necessarily Whose Pattern of Life

Massive Data Sets for
Anomaly / Change Detection

Indications and Warnings

Massive Data Aggregation for Machine Analytics, **Baselining,** and **Trend Analysis**

Non-Attributable Aggregate Behavior
- Determine Avg Traffic Speed by Tracking Cell Movement
- Determine the Sentiment of a Town, City, Region, Country

Non-Attributable

Population Centric

UNITED STATES ARMY INTELLIGENCE AND SECURITY COMMAND

Simple Search, Graph & Document Viewers
(Entity Search, Entity Resolution, Social Analysis)

- All Searches Are Against ***All the Data*** (Structured and Unstructured)
  - Unlike Current Systems, *Don't Need to Filter Data*
- Search Results Returned at Internet Speed
- "Chinese Menu" Provides Rapid Discovery of Unknown Connections
- Immediately Enter Analysis (Seconds vs Minutes/Hours)
- Click to Explore Connections for Relevance *(Snippets)*
- De-clutters Visual Display, Without Filtering
- Adding Context Leads to "Precision Search"

- Associative Memory Index

Widget Interaction: Patterns of Life
(Widgets: Time Wheel, Histogram, Map)

- **Simple Functions Combine for Powerful Analytic Capability**
- **Decouples Functionality for Widget Reuse *(Pull Together like LEGOs)***
- **Ease of Widget Creation**
    - **Lightweight Software Coding (< 100 Lines of Java Script)**
    - **Open Commercial Standards**
    - **Users Can Build Rapidly (Secure Container)**
    - ***Rapid Response* to User Needs and User Feedback**
    - ***Harnesses the Innovation of our Users***

Contextual Search / N-Gram
(Enhanced Searching)

- Aggressively Index Every Word
- Data Self-Organizes for Analysts to See Context of Terms
- Internet Speed of Response
- Analysts Form Relevant Queries in Seconds from a Cold Start
- Enables *Precision Search*
- Foundational Index - Building Block for other Widgets + Micro-Analytics
    - MSR Example: Map the Documents
    - Name Variant Example: Select Common Mis-Spellings to Check ALL

Community of Interest (COI)
(Enhanced Searching, Alerting, Social Networking)

- Use Cloud Analytics Not Just on Red Threat Data, but to Manage BLUE
  - Leverage Knowledge on How the Network is Used
  - Exploit User Profiles and Expertise Across the Enterprise

- Better, Faster Analysis:
  - Link with Other Analysts Working Same Problem *(Auto-find SMEs)*
  - Give Recommendations Based on Users' Expertise

- Training: Enables Supervisor Oversight and Management of Analysts
  - Detect Weak Analytic Skills *(Avoid "Analytic ADD")*
  - Prevent Too Many Analysts from Working Same Problem

- Detect Anomalous Activity *(Counter Insider Threat)*
  - Users Accessing Data Beyond Assigned Function
  - Excessive Downloading *(WikiLeaks)*

CI-HUMINT / CHAMPION
(Integration of “Legacy Systems”)

- Demonstrate the Flexibility of the Cloud to Integrate Legacy “Boxes”
  - Widgetize Functionality: 1/2 Month
  - Complete Integration into Cloud: 2-1/2 Months
  - Total Effort to Integrate Legacy → Cloud: 3 Months
    - *Compare with Traditional 18-24 Months to Deploy, and Benefit from Less HW / SW / Support Personnel / Footprint*

- Decouple Proprietary Services and Reuse Existing Services
- Focus Development Resources on New Capabilities

Cloud-to-Cloud
(Forming the Ubiquitous Cloud)

- Demonstrate Cross-Cloud Functionality
- Allows Data Stewards to Manage Data
    - Others Can *Leverage, not REPLICATE,* Vast Amounts of Data
- Access Others' Data, Services and Hardware at No Added Cost
- Reduced Bandwidth Required (Only Pass Results, not Underlying Data)
- *Empowered by an IC-common, Non-Proprietary, Open Architecture*

Cloud-enabled Advanced Analytics

- Exponential Improvement
  - Reduces Time for Analysis (Hours → Minutes)
  - Enables Fast Precision Search of Huge Data
  - Every Query Searches ALL Data; No Data Filtering Needed

- Power of IC-Common Cloud HW, SW, Visualization
  - Shared Investments Accelerate Progress
  - Common Standards Allow Widgets to Communicate, Even If Built Separately
  - Development & Fielding Cycle Reduced from Years to Months: Meet Needs Rapidly

Responsive, Agile Innovation to Transform Army Intel

UNITED STATES ARMY
INTELLIGENCE AND SECURITY COMMAND

Lean

Lean and Kanban

Agile

Michael Cox

Vice President and Senior Consultant

NetObjectives, Inc.

michael.cox@netobjectives.com

"What is all this Agile stuff about, anyway?"

Lean and Kanban

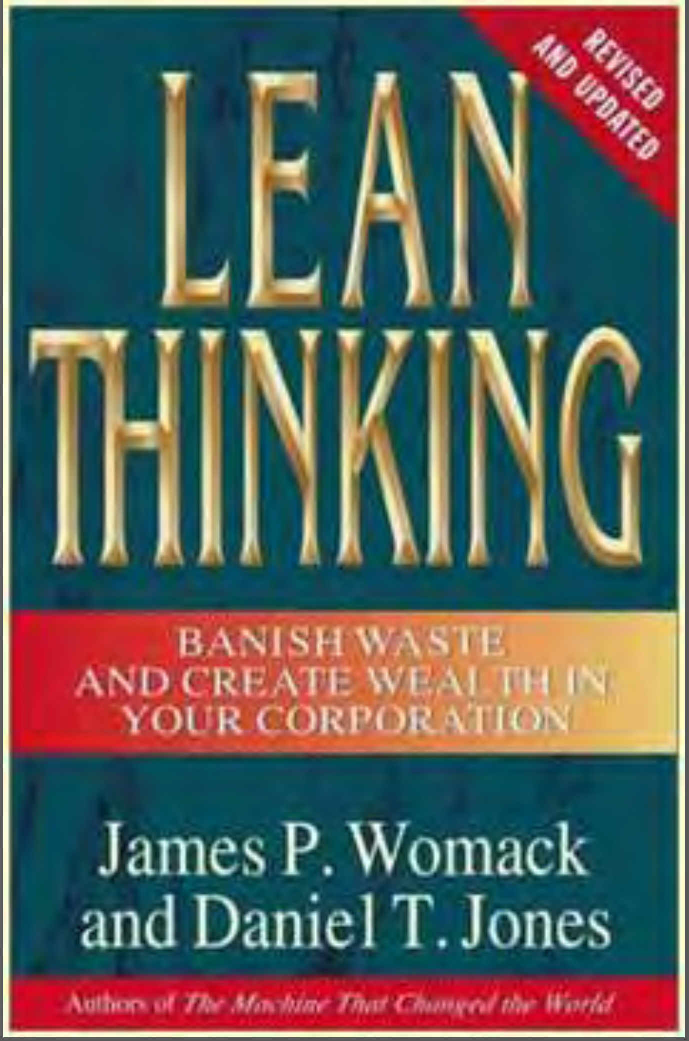
How do they compliment each other?

How do you use them?

Why does it work?

Lean

- Value from the customers perspective
- Identify and eliminate waste – non value added activities
- Flow of work at customer demand
- Continuous improvement


REVISED AND UPDATED
LEAN THINKING
BANISH WASTE AND CREATE WEALTH IN YOUR CORPORATION
James P. Womack and Daniel T. Jones
Authors of The Machine That Changed the World

Kanban

A management discipline.
A constant exercise of matching demand with supply, to deliver the right thing at the right time.

See also: Visibility, Prioritization, WIP limits, Pull

Agile

Agile is a method that features rapid delivery of functional product iterations

Relies on immediate customer feedback

Allows for evolving understanding of system

Agile

Agile is about

Business Iterations *not* Development Cycles

Agility

Predictability

of Business Value

Realization

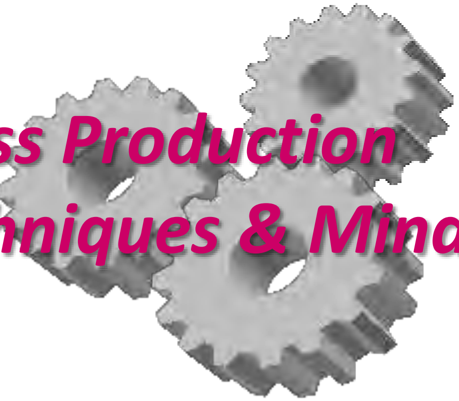
“Where did this stuff come from?”

How Did We Get Into This Spot?

- Tremendous rise in the standard of living the past 100 years in all developed countries
- Rise was largely driven by productivity improvements
  - Agricultural up 3 to 5% a year since 1900
    - 50% of workforce in 1900, < 2% today, more production
  - Production up by 3% a year since Depression
    - 35% of workforce in 1940, < 15% today, 100x output rise

Basis has been the Invention and Widespread Adoption of Mass Production Techniques

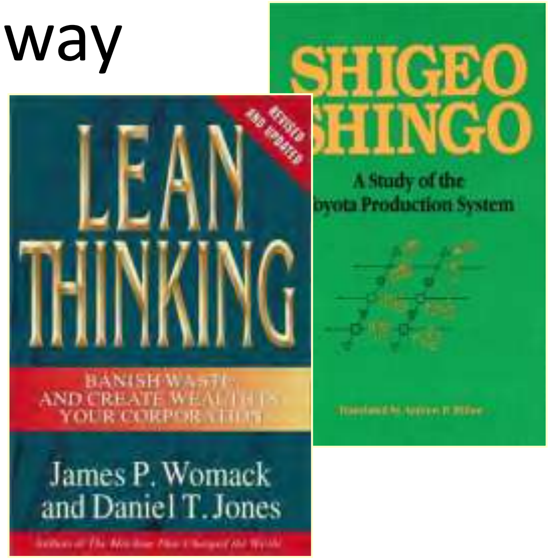
How Did We Get Into This Spot?



Mass Production Techniques & Mindset

Thought Basis for Current Management Practices

- Managing via hierarchy, command and control
- Scientific management – the one best way
- Economies of scale
- Batch production


LEAN THINKING
REVISED AND UPDATED
BANISH WASTE AND CREATE WEALTH IN YOUR CORPORATION
James P. Womack and Daniel T. Jones
SHIGEO SHINGO
A Study of the
oyota Production System

Lean Principles have generated Lean Practices

How Did We Get Into This Spot?

- Mass production management techniques in systems and software development have largely failed
  - Documentation = Understanding
  - The right tasks, correct pressure - force it to happen
  - "If they would freeze requirements, we would be fine"
  - "Heroes" called in when program is in real trouble
- A dissatisfied customer community has imposed more controls and rigidity
- Contractors countered with rigid contracts and change orders to batter the customer with cost and schedule
- Product owners were not involved until too late

we are always working with uncertainty

Lean and Kanban help us deal with uncertainty

The result is agility

Lean suggests limit TIME between steps

time

Kanban suggests limit # of items being worked on in each step

size of queue

Understanding Lean

1. Value from the Customer's Perspective
2. Value stream
3. Flow
4. Pull
5. Perfection

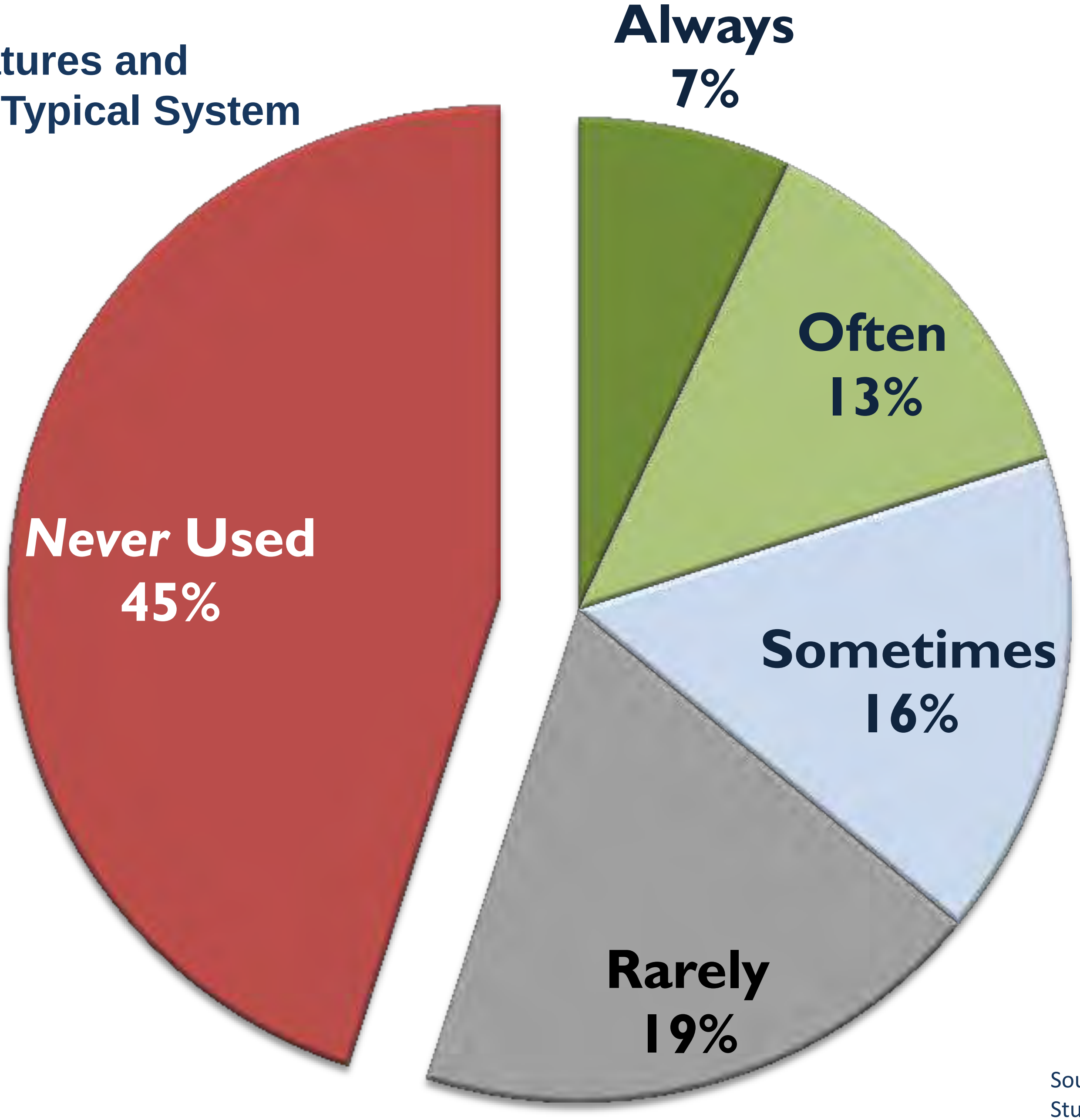
- Define the value
- See the value stream
- Flow and where value comes from
- JIT
- Cycle time
- Reduce waste

thinking points

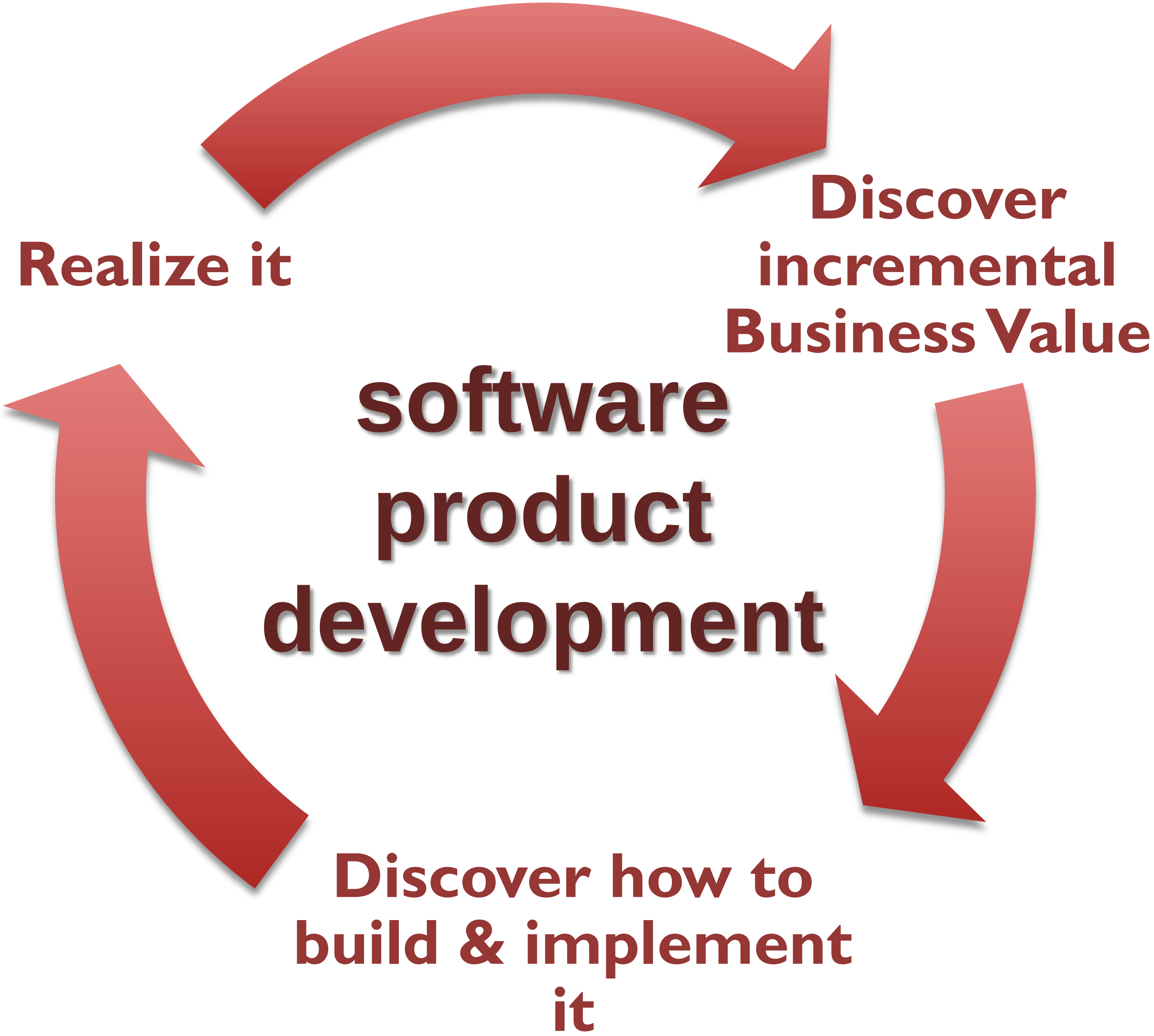
“You cannot build the right thing
if you have not discovered it first!”

The product owner must own the definition of value!

Usage of Features and Functions in Typical System



Source: Standish Group
Study of 2000 projects at
1000 companies

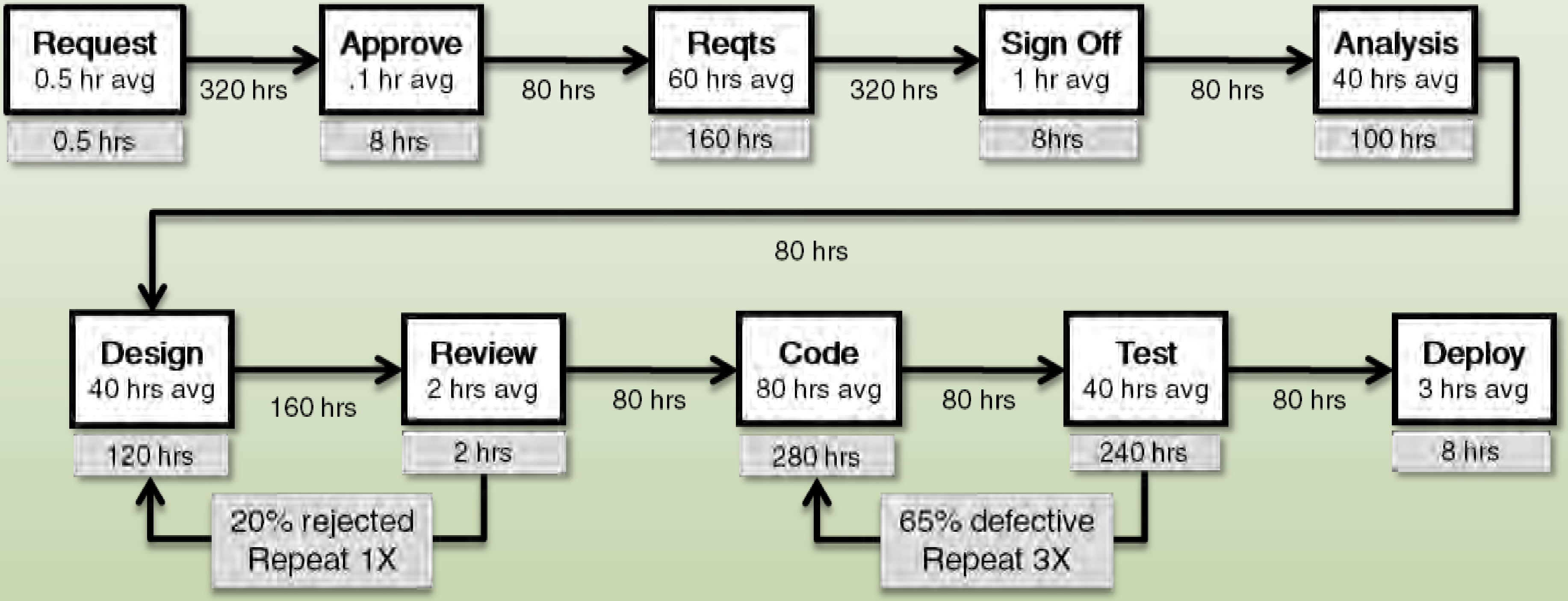


software product development

Realize it

Discover incremental Business Value

Discover how to build & implement it



Request
0.5 hr avg
0.5 hrs

320 hrs

Approve
.1 hr avg
8 hrs

80 hrs

Reqts
60 hrs avg
160 hrs

320 hrs

Sign Off
1 hr avg
8hrs

80 hrs

Analysis
40 hrs avg
100 hrs

80 hrs

Design
40 hrs avg
120 hrs

160 hrs

Review
2 hrs avg
2 hrs

20% rejected
Repeat 1X

80 hrs

Code
80 hrs avg
280 hrs

80 hrs

Test
40 hrs avg
240 hrs

65% defective
Repeat 3X

80 hrs

Deploy
3 hrs avg
8 hrs

visualize the **entire** value stream

The value stream

- Continuous flow of valuable work and features into deployment
- Includes *everybody* from the customer to operations and support engineers, and *not just development*

visualize the **entire** value stream

Flow

Focus on TIME

Large batches create delay and waste

while

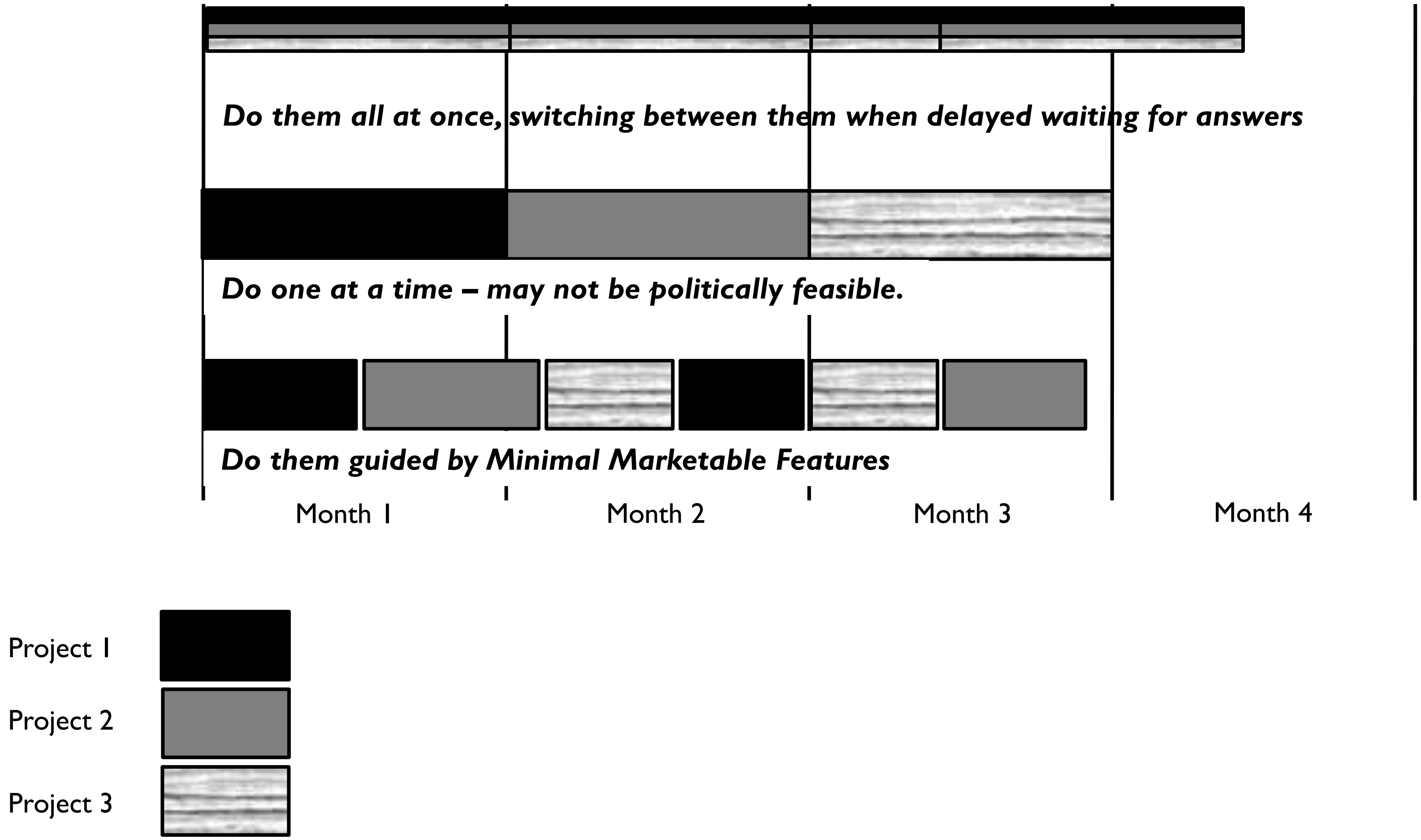
Small batches create incremental value

Optimize the whole!

Task-Switching and Schedules

LEAN THINKING

Three ways to do three projects



Do them all at once, switching between them when delayed waiting for answers

Do one at a time – may not be politically feasible.

Do them guided by Minimal Marketable Features

Product Development for the Lean Enterprise by Michael Kennedy. Oaklea Press. 2003

DELAY IS

finding

redoing

reworking

waiting

hand-offs
bottlenecks
information delay
untested code
unread requirements
transaction related
coordination related

Cycle Time is Key!

Requirements …

Decay and

Lose Value

over time

Requirements

are not fully understood even after a formal sign-off

Requirements

change often
during long development cycles

Requirements

piled on
poorly prioritized
long delivery cycles

Pull

The work enters a queue.

When someone needs new work, they pull from the queue.

The work goes through a number of stages. When the work is done in a stage, it flows down to the next stage.

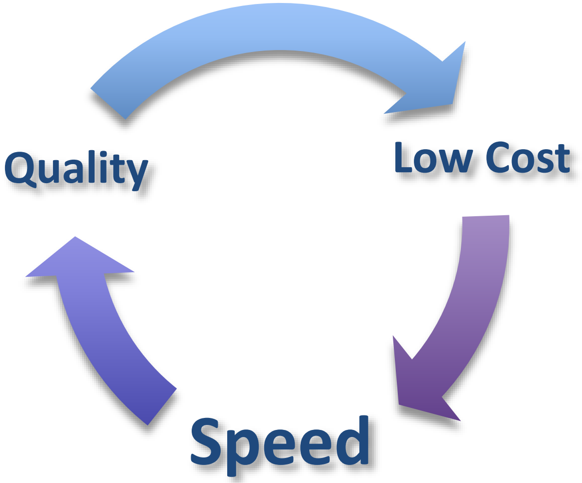
Until it is done.

Short Cycle Time

Principles of Lean Software Development

- Optimize the Whole
- Eliminate Waste
- Build Quality In
- Deliver Fast

- Defer Commitment
- Create Knowledge
- Empower People



kanban

improves quality and *lowers* cost

by ***eliminating delays***

by **managing WIP**

Kanban for Systems and Software

Workflows can be seen and managed

You can divide the work into small value adding increments

It is possible to develop value-adding increment in a continuous flow, from requirement to deployment

Kanban for Systems and Software

Limit Work in Process (WIP)

Pull value through

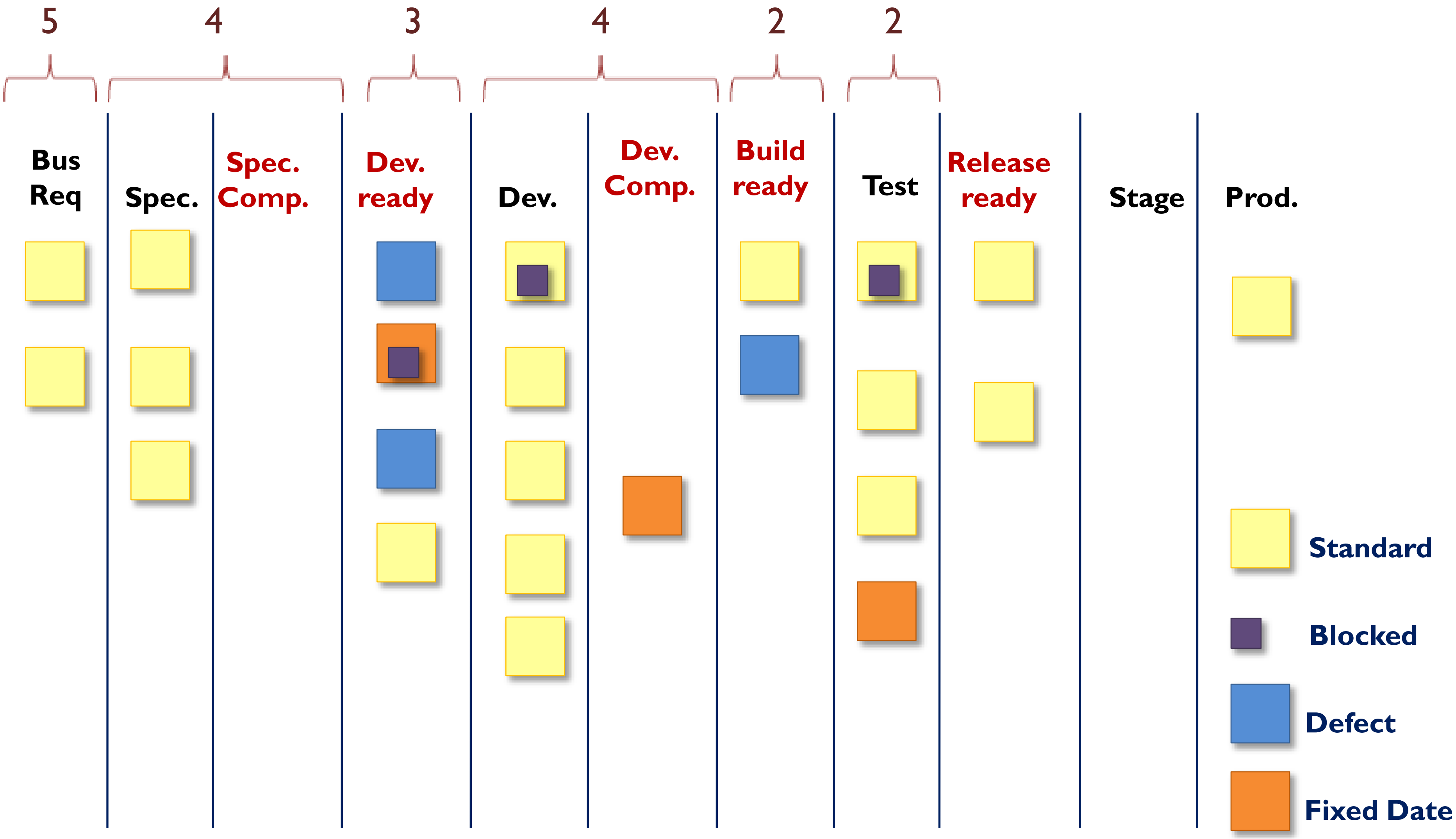
Make it visible

Increase throughput

Prioritized Backlog

Quality is built in

Team continuously monitor and improve



5 4 3 4 2 2

Bus Req

Spec.

Spec. Comp.

Dev. ready

Dev.

Dev. Comp.

Build ready

Test

Release ready

Stage

Prod.

🟨 Standard

🟪 Blocked

🟦 Defect

🟧 Fixed Date

Courtesy Olav Maassen QNH

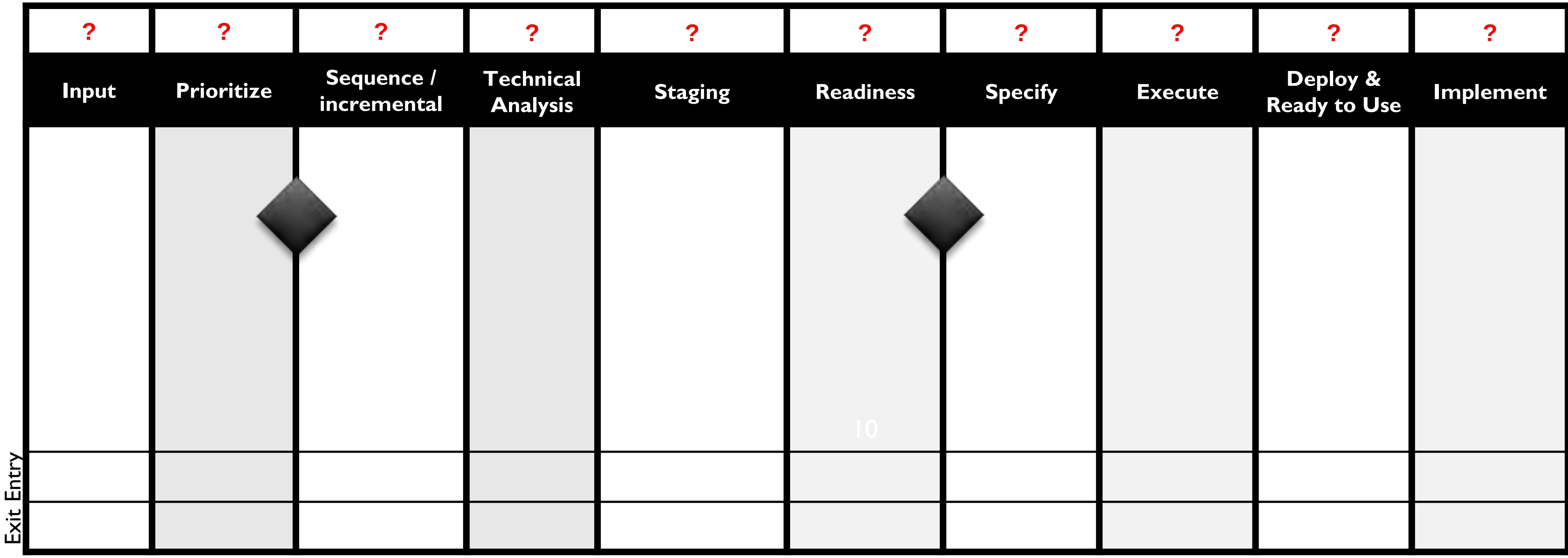
design the kanban board

make all work visible

Kanban boards reflect

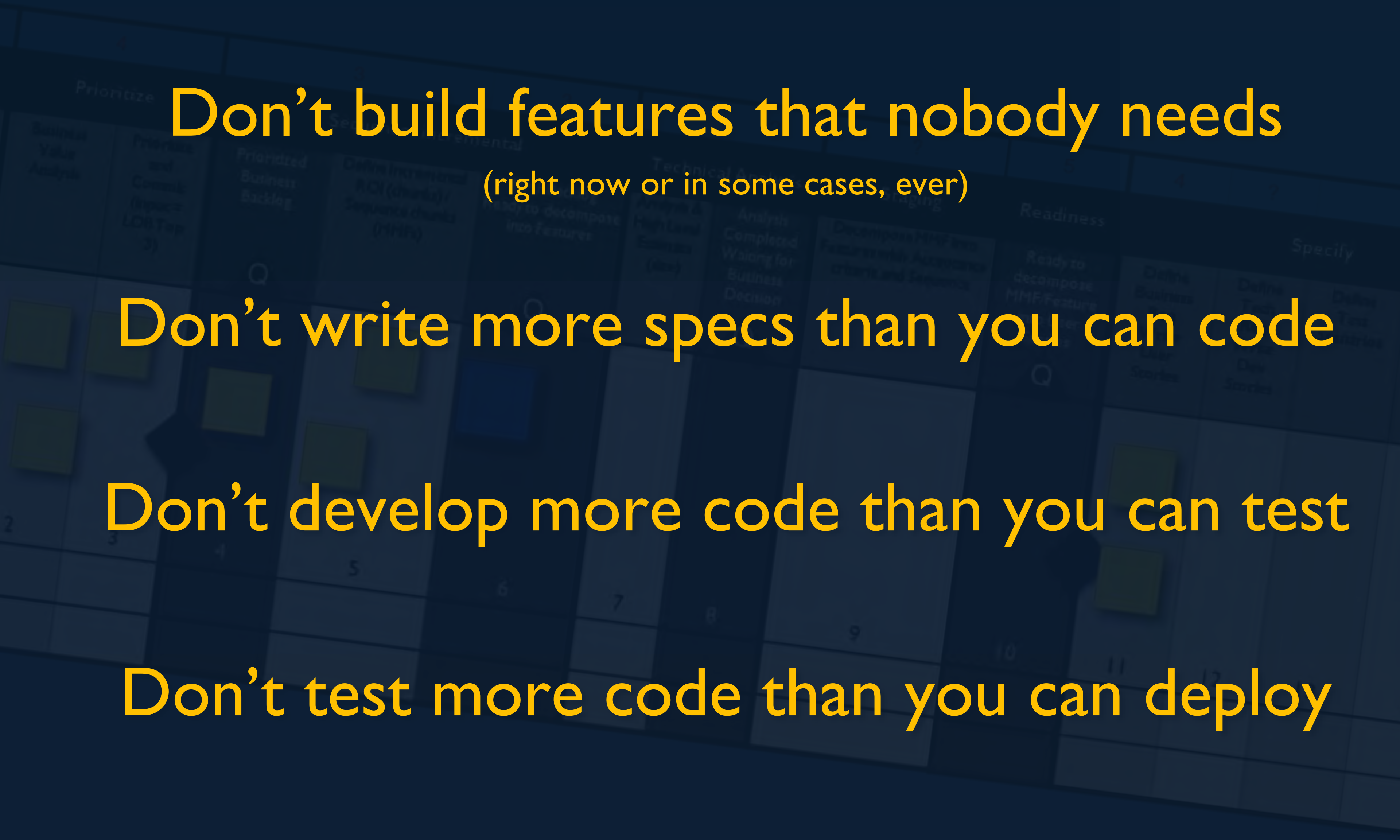
- Priority
- WIP
- Process

Business Value Kanban



Business Discovery | *Business Delivery*

	Input	Prioritize	Sequence / incremental	Technical Analysis	Staging	Readiness	Specify	Execute	Deploy & Ready to Use	Implement
	?	?	?	?	?	?	?	?	?	?
						10				
Entry										
Exit										



Don’t build features that nobody needs

(right now or in some cases, ever)

Don’t write more specs than you can code

Don’t develop more code than you can test

Don’t test more code than you can deploy

Kanban Success

Focus on Quality

Reduce WIP

Balance demand against throughput

Prioritize

lean thinking

Courtesy David Anderson

throughput

utilization

Stop Starting and Start Finishing

David Anderson

Pull

The work enters a queue.

When someone needs new work, they pull from the queue.

The work goes through a number of stages. When the work is done in a stage, it flows down to the next stage.

Until it is done.

WIP Limit…

Governs the *maximum* number of work items that can be in that state at any instant.

Below its limit:

Receive a work item from upstream

At its limit:

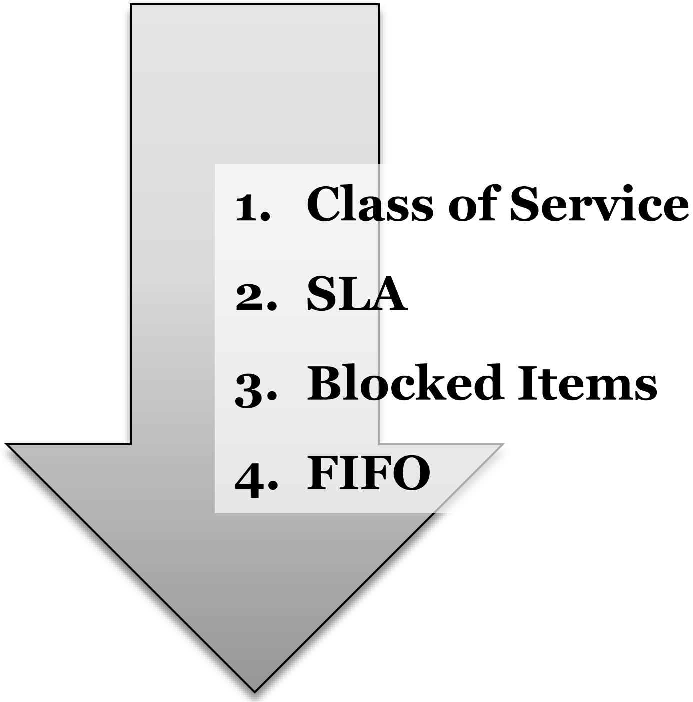
Wait for one of its own items to be completed and flowed downstream

In Knowledge Work, complexity grows exponentially with WIP

Classes of Service

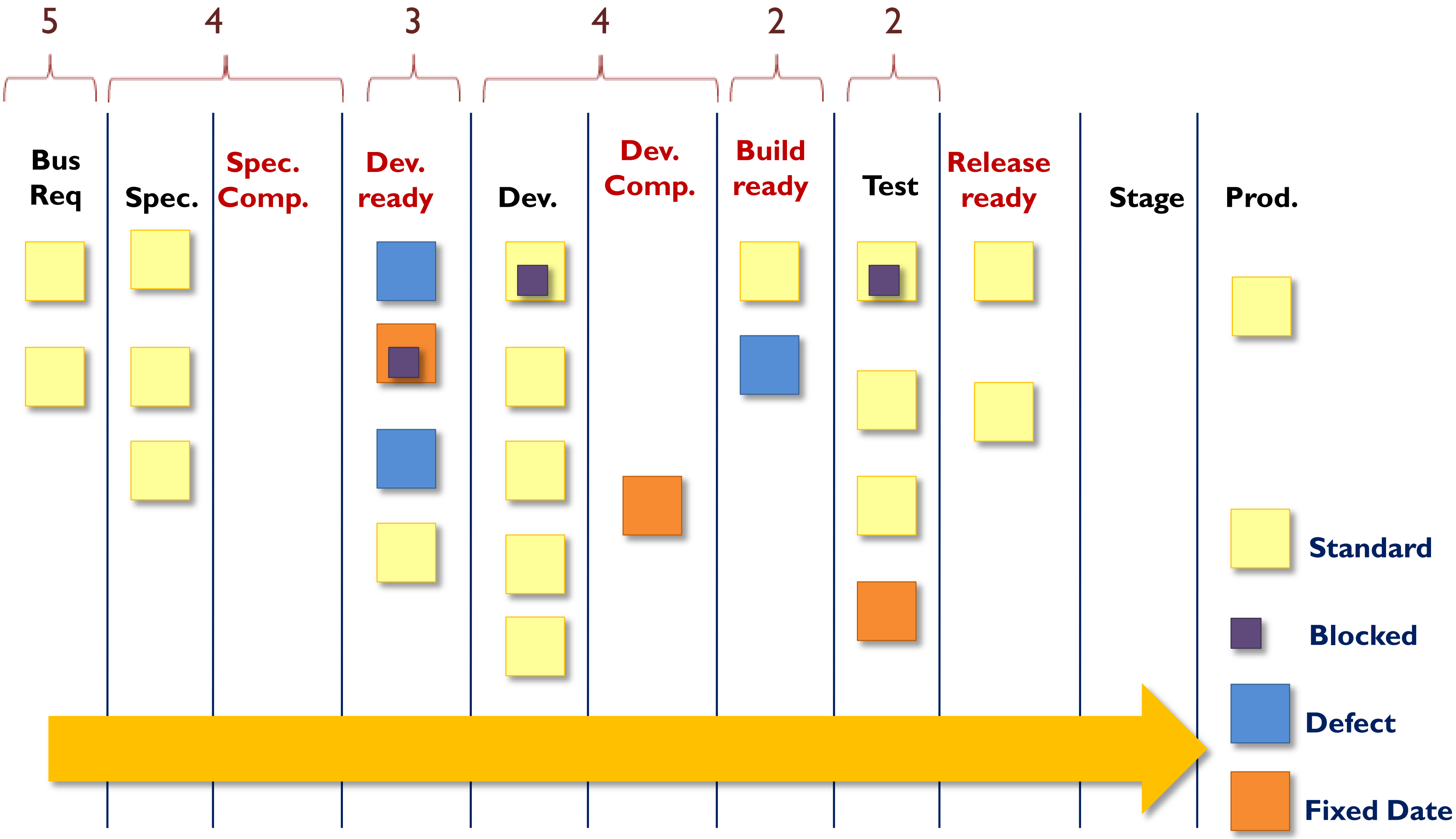
- 🟪 Expedite
- 🟨 Specific Delivery Date
- 🟩 Standard
  Maintenance or
  Break-Fix Work
- 🟦 Standard
  New or
  Value-Added Work
- ⬜ Development Story
- 🟧 Outside Impact
- 🟪 Impediment
- 🔴 Red Flag Issue

Policies & SLAs



1. **Class of Service**
2. **SLA**
3. **Blocked Items**
4. **FIFO**

Direct the team
in the **priority** of
processing work items



5 4 3 4 2 2

Bus Req

Spec.

Spec. Comp.

Dev. ready

Dev.

Dev. Comp.

Build ready

Test

Release ready

Stage

Prod.

🟨 Standard

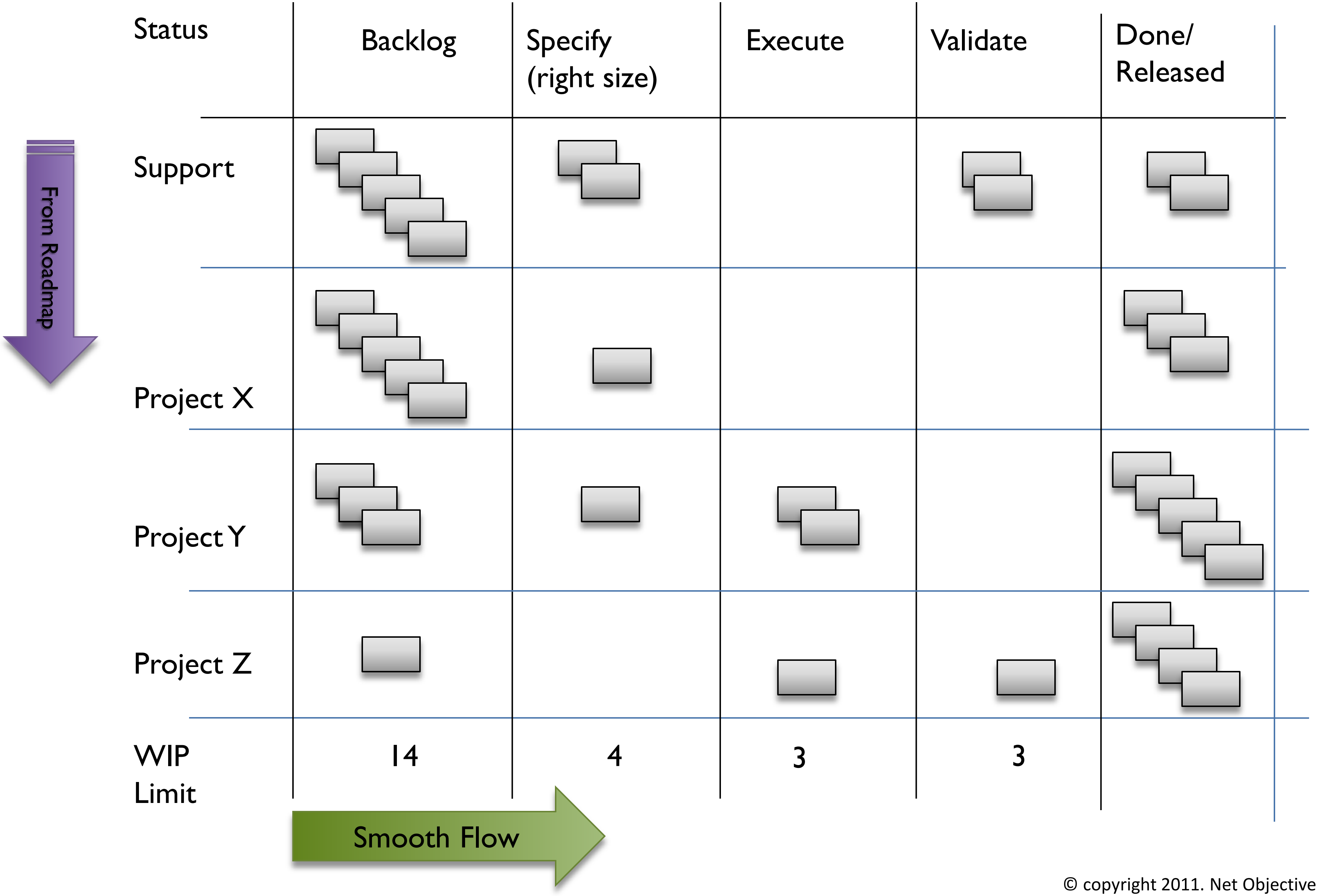
🟪 Blocked

🟦 Defect

🟧 Fixed Date

Courtesy Olav Maassen QNH

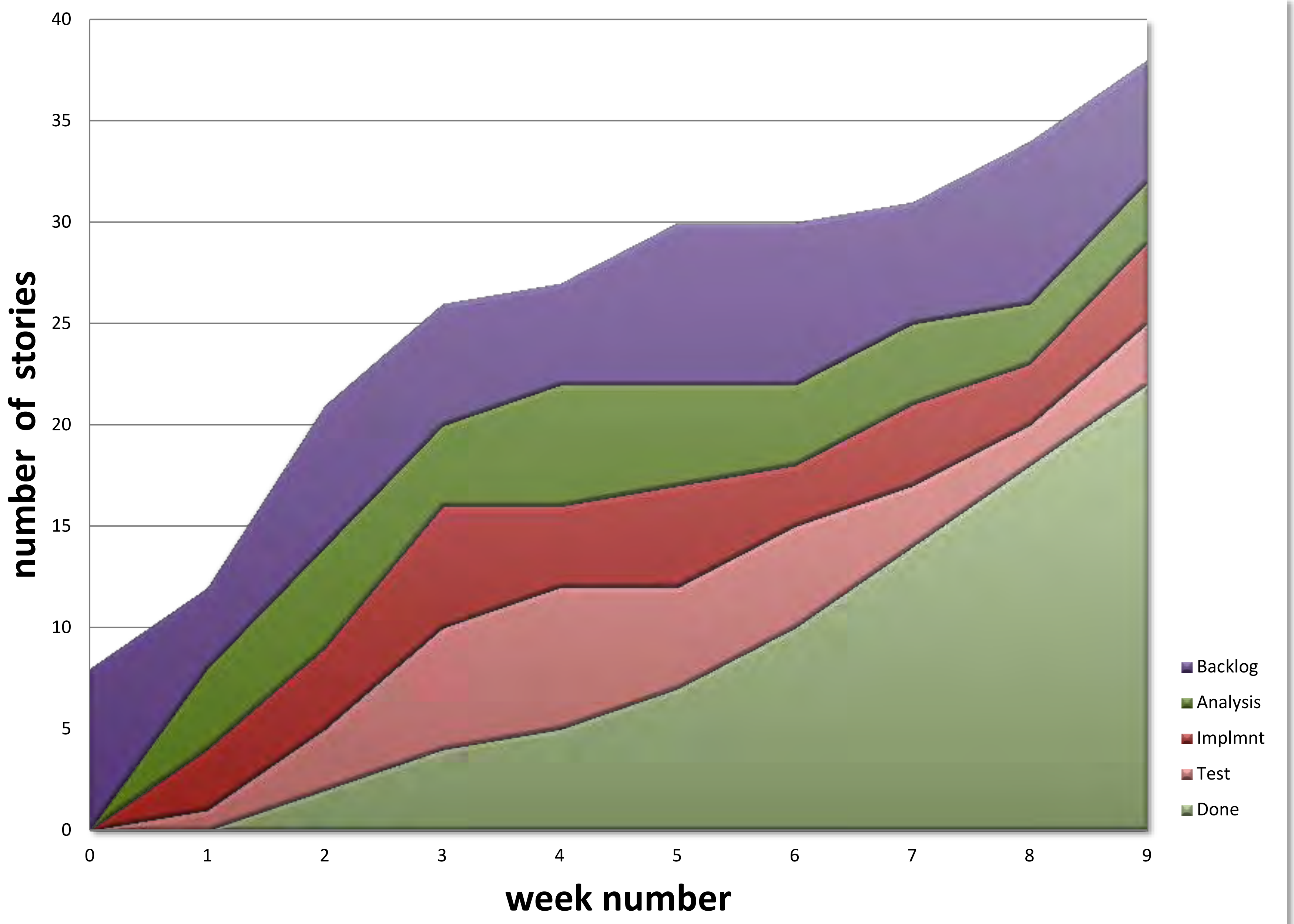
Monitoring flow: Kanban for portfolio



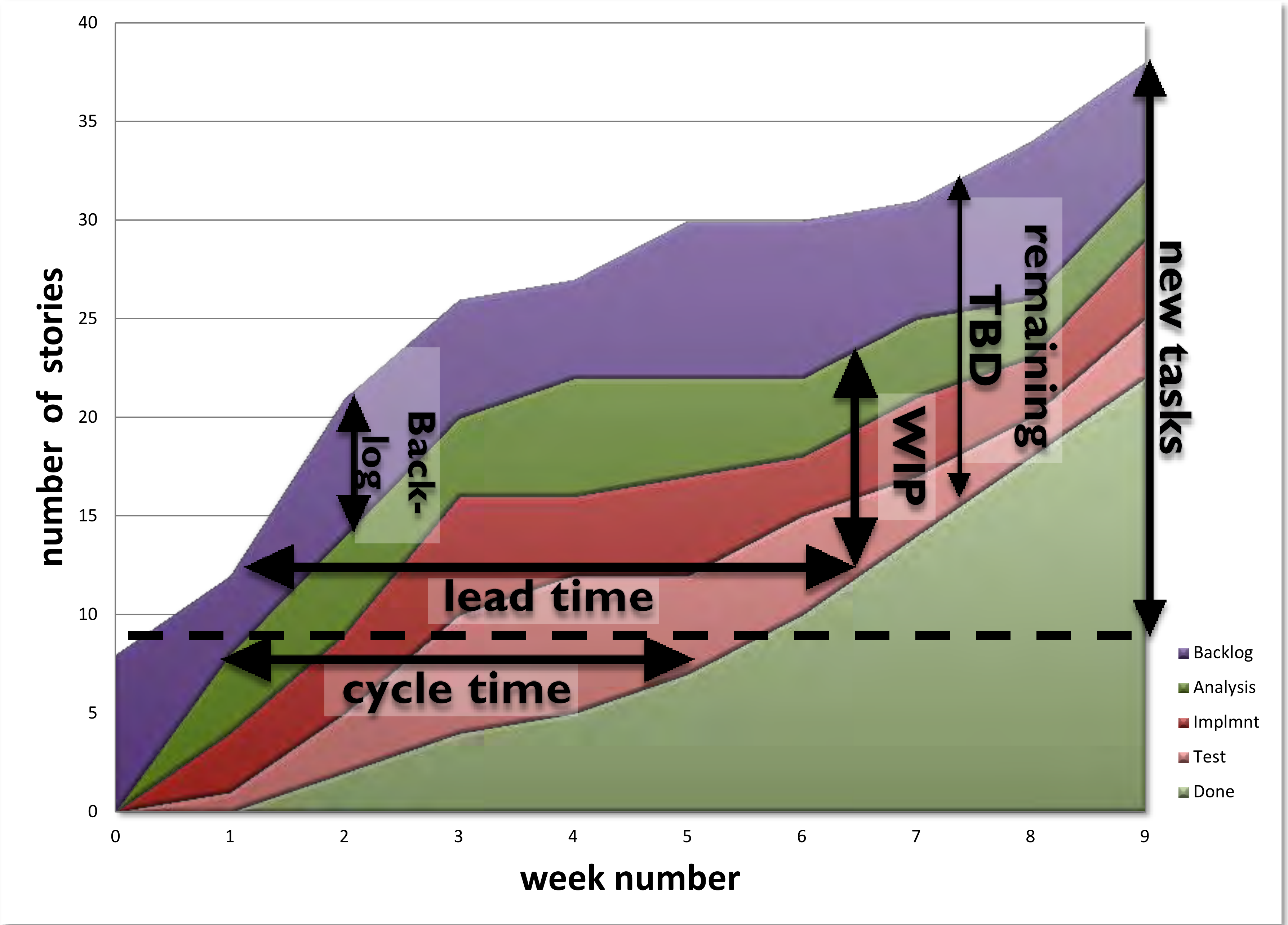
Status	Backlog	Specify (right size)	Execute	Validate	Done/ Released
Support	▭▭▭▭▭	▭▭		▭▭	▭▭
Project X	▭▭▭▭▭	▭			▭▭▭
Project Y	▭▭▭	▭	▭▭		▭▭▭▭▭
Project Z	▭		▭	▭	▭▭▭▭
WIP Limit	14	4	3	3	

From Roadmap

Smooth Flow



cumulative flow diagram



cumulative flow diagram

always focus on

WHAT SHOULD WE FINISH NEXT?

Kanban Stand-up

GUIDELINES

Who Talks?

- Only Team members moving stickies across the board!

Do This

- Start from the **right**
- Work by the **highest priority**
- Pay attention to:
  - Oldest
  - Blocked
  - Class of Service
  - SLA in jeopardy
- Ask
  - Do we have a bottleneck (congestion or gaps in the queues)?
  - Do we have a "blocker" not dealt with?
  - Are we keeping to our WIP limits?
  - Are priorities clear?

When done

- Update the board
- Remove done items from the board

Getting started with kanban

- **Agree to goals**
- **Map the value stream**
- **Define a set of work item types**
- **Meet with external stakeholders**
- **Create board for tracking**
- **Agree to standup**
- **Agree to operational review**
- **Educate the team**
- **Start doing it**

David Anderson. XTC, London 2009, October

Kanban

What you will see:

- Queues start backing up immediately following any blockage
- Predictable consequences
- The entire board will slow down as a result of flow issues
- Teams see issues right away and act together to fix them

Lean and Kanban

Lean is the theory

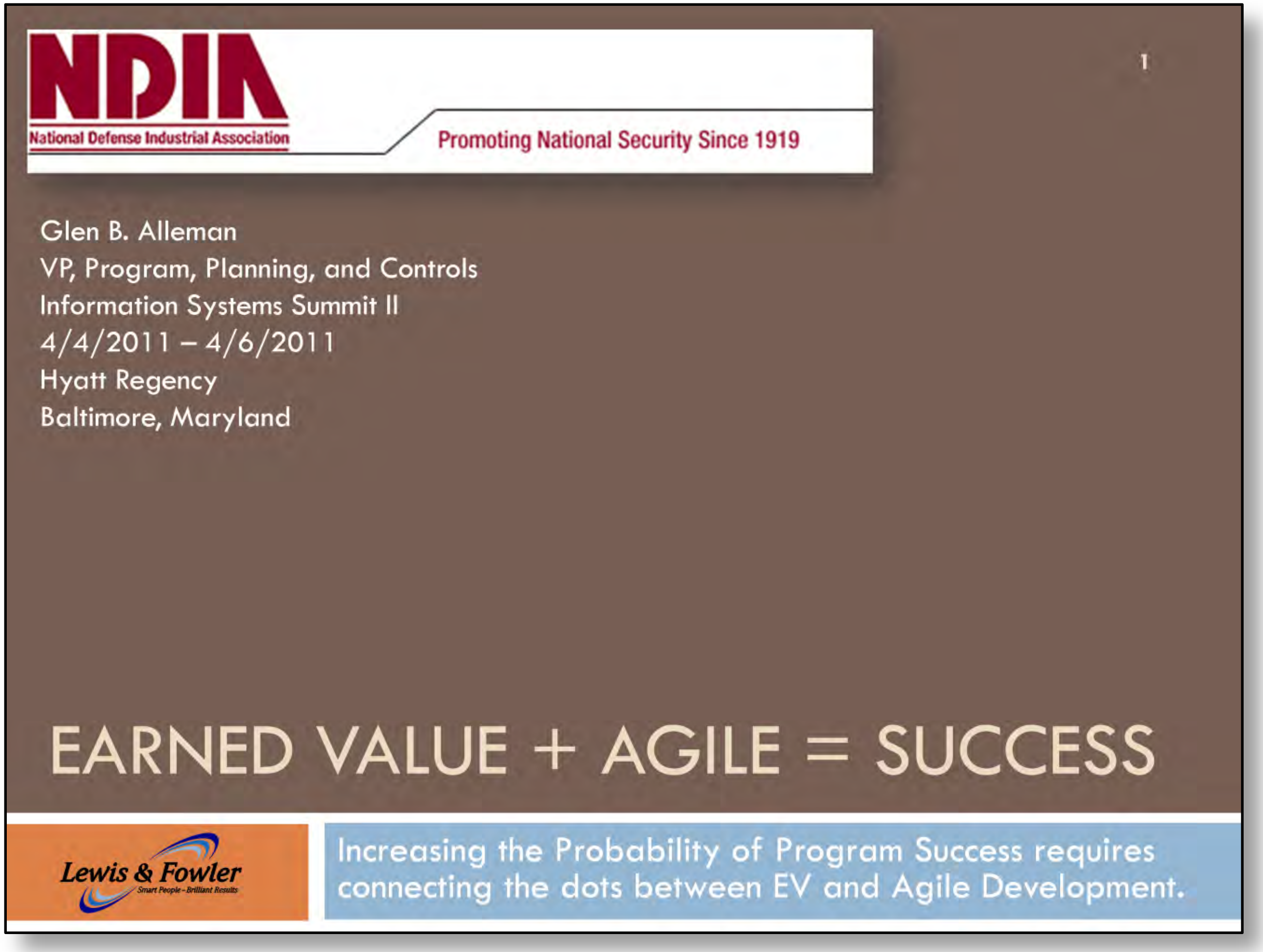
Kanban is the approach

Agile is the result

Questions

?

Earned Value + Agile = Success
Glen B. Alleman, VP Program Controls, Lewis & Fowler
NDIA Information Systems Summit II
4/4/2011 – 4/6/2011
Hyatt Regency, Baltimore, Maryland



NDIA
National Defense Industrial Association
Promoting National Security Since 1919

1

Glen B. Alleman
VP, Program, Planning, and Controls
Information Systems Summit II
4/4/2011 – 4/6/2011
Hyatt Regency
Baltimore, Maryland

EARNED VALUE + AGILE = SUCCESS

Lewis & Fowler
Smart People – Brilliant Results

Increasing the Probability of Program Success requires connecting the dots between EV and Agile Development.

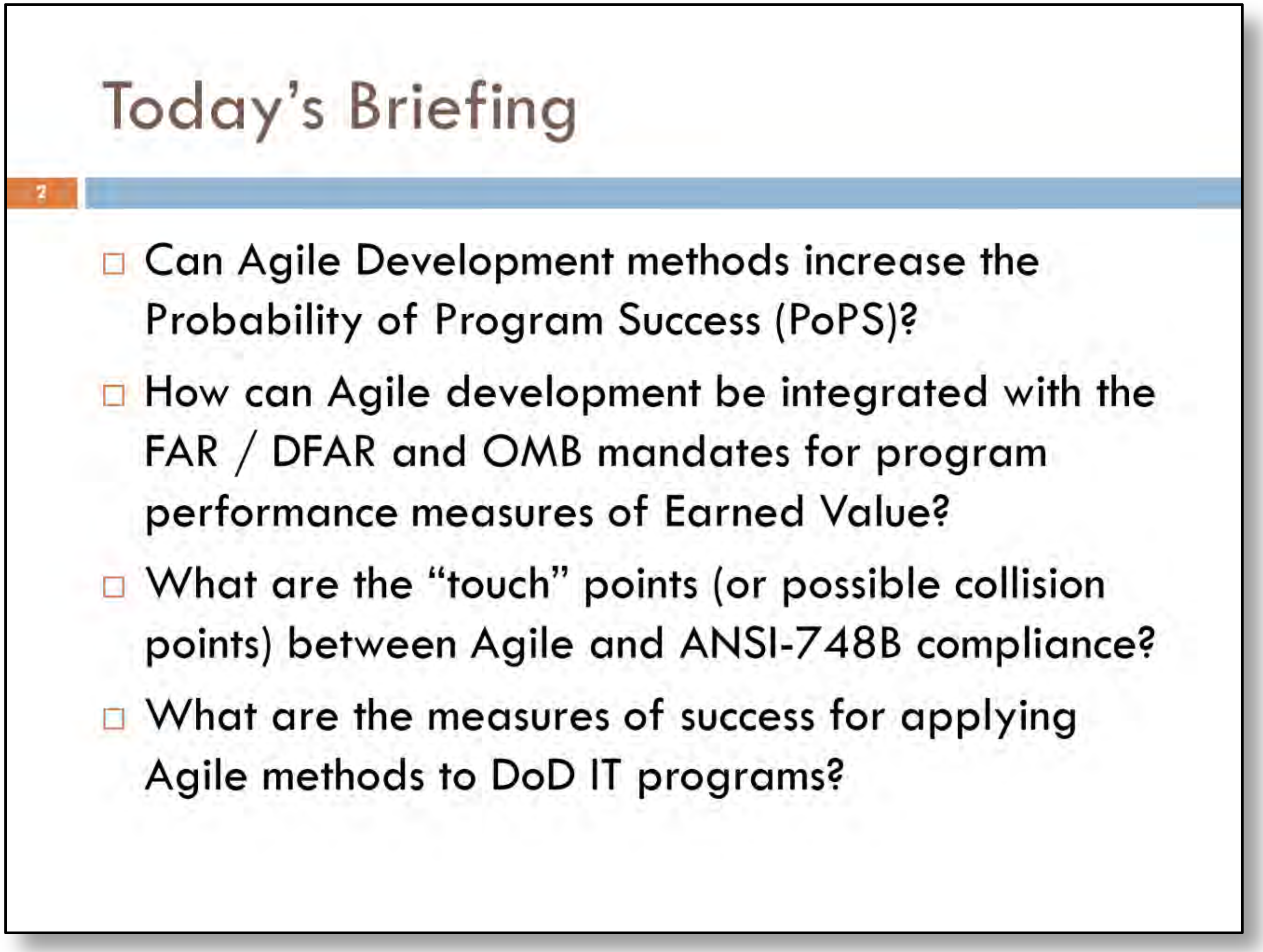
Thank you for having me this morning. You've heard many speakers address way of developing software using agile development methods.

That is not the topic of this briefing.

I'm going to introduce a parallel topic to the development of software using agile methods.

This topic starts and ends with the requirement – a Federal Acquisition Regulation requirements – for the application of Earned Value Management for programs greater than $20M and for the use of a DCMA validated system for programs greater than $50M.

We'll see the sources of this guidance in a moment. But no matter what the guidance says, how it is applied – or not applied – I'm going to try and convince you that Earned Value Management is a good thing in the context of Agile Software Development and the directive that comes form the NDAA 2010, Section 804.



Today's Briefing

- □ Can Agile Development methods increase the Probability of Program Success (PoPS)?
- □ How can Agile development be integrated with the FAR / DFAR and OMB mandates for program performance measures of Earned Value?
- □ What are the "touch" points (or possible collision points) between Agile and ANSI-748B compliance?
- □ What are the measures of success for applying Agile methods to DoD IT programs?

Before any of the current "agile" development methods were around, Earned Value Management provided information for planning and controlling complex projects by measuring how much "value' was produced for a given cost in a period of time. With the connection to the Business Value in agile, both technical performance and business performance can be used to guide the performance of an enterprise IT project.

The concept of Probability of Program Success is applied to other DoD Acquisition process in the Air Force, Army, and Navy. It asks and answers the question "what are the key performance parameters (KPP) for the success of the program?"

While agile's contribution to the development of software is the topic of many of the speaker, I'd like to introduce the notion that projects and programs in the US Department of Defense are still subject to the Federal Acquisition Regulation (FAR) and Defense Federal Acquisition Regulation (DFAR) once the program has reached a predefined dollar value.

At some point in the IT procurement process, it is likely a DoD IT program will cross that threshold.

What Do We Mean When We Say Agile?

Being able to turn inside the loop of unfolding events. †

Pembroke Welsh Corgi in a goat herding competition, Boulder County Fair, Longmont Colorado
Chubby body, short legs, not "lean," but able to turn "inside the loop" of the goat = "agile?"

3

† Dr. Ashton Carter, Under Secretary of Defense for Acquisition, Technology and Logistics, Sep/Oct, 2010 *Defense AT&L*

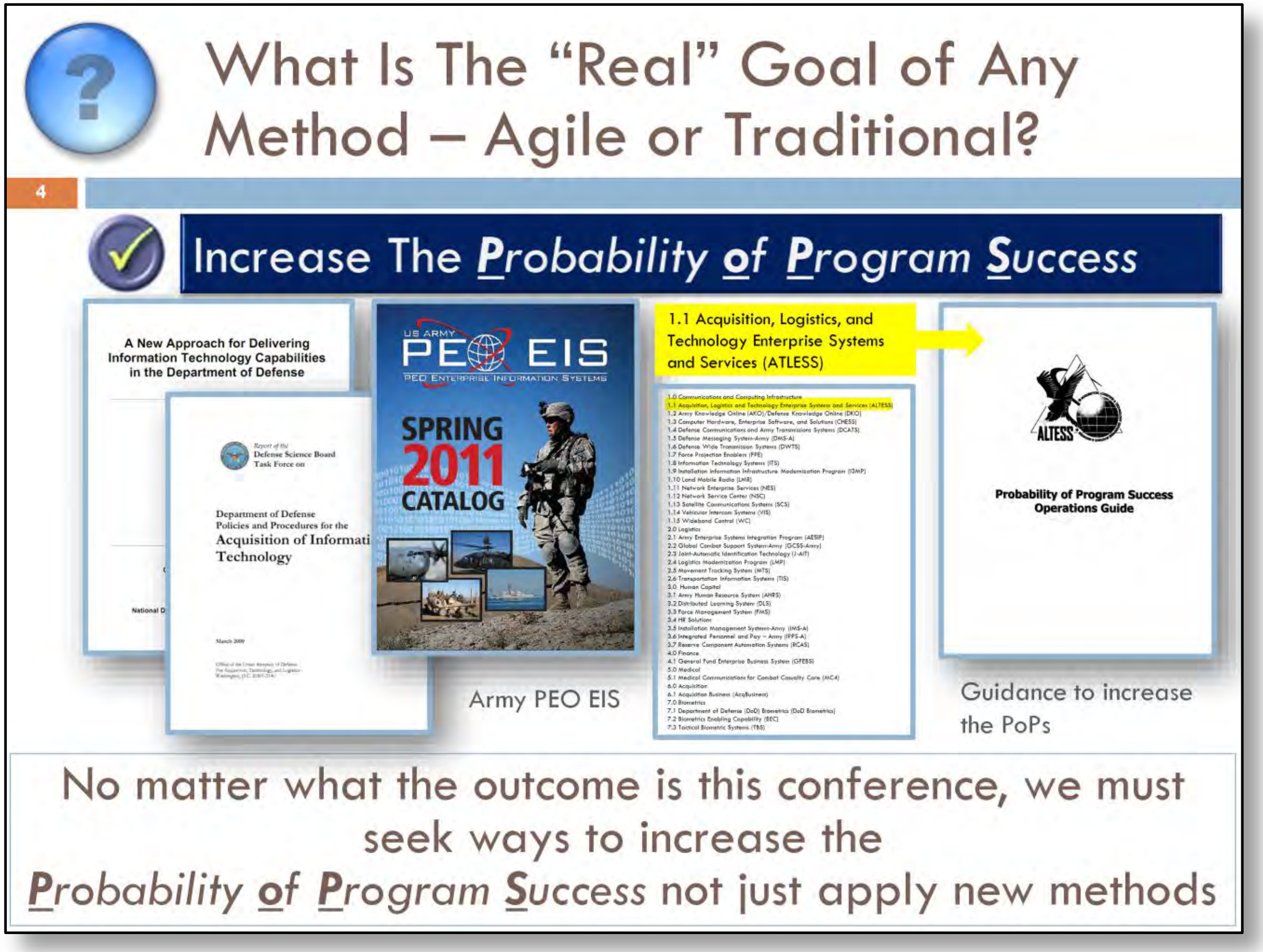
There are lots of definitions of agile. Most come from the software development world. But let's have a definition that is meaningful to the problem at hand. That problem is defined in NDAA Section 804's instructions. If we haven't heard of NDAA Section 804, it's the National Defense Authorization Act 2010, Section 804. we'll see the details in a bit, but for now Section 804 says:

- SEC. 804. IMPLEMENTATION OF NEW ACQUISITION PROCESS FOR INFORMATION TECHNOLOGY SYSTEMS.
- The Secretary of Defense shall develop and implement a new acquisition process for information technology systems. The acquisition process developed and implemented pursuant to this subsection shall, to the extent determined appropriate by the Secretary
- Be based on the recommendations in chapter 6 of the March 2009 report of the Defense Science Board Task Force on Department of Defense Policies and Procedures for the Acquisition of Information Technology; and

(2) be designed to include—

- (A) early and continual involvement of the user;
- (B) multiple, rapidly executed increments or releases of capability;
- (C) early, successive prototyping to support an evolutionary approach; and
- (D) a modular, open-systems approach.

The last four phrases should be sound familiar to any of you practicing agile software development.



What Is The "Real" Goal of Any Method – Agile or Traditional?

4

Increase The Probability of Program Success

A New Approach for Delivering Information Technology Capabilities in the Department of Defense

Report of the Defense Science Board Task Force on

Department of Defense Policies and Procedures for the Acquisition of Informati Technology

March 2009

US ARMY PEO EIS
PEO ENTERPRISE INFORMATION SYSTEMS
SPRING 2011 CATALOG

Army PEO EIS

1.1 Acquisition, Logistics, and Technology Enterprise Systems and Services (ATLESS)

1.0 Communications and Computing Infrastructure
1.1 Acquisition, Logistics and Technology Enterprise Systems and Services (ALTESS)
1.2 Army Knowledge Online (AKO)/Defense Knowledge Online (DKO)
1.3 Computer Hardware, Enterprise Software, and Solutions (CHESS)
1.4 Defense Communications and Army Transmissions Systems (DCATS)
1.5 Defense Messaging System-Army (DMS-A)
1.6 Defense Wide Transmission Systems (DWTS)
1.7 Force Projection Enablers (FPE)
1.8 Information Technology Systems (ITS)
1.9 Installation Information Infrastructure Modernization Program (I3MP)
1.10 Land Mobile Radio (LMR)
1.11 Network Enterprise Services (NES)
1.12 Network Service Center (NSC)
1.13 Satellite Communications Systems (SCS)
1.14 Vehicular Intercom Systems (VIS)
1.15 Wideband Control (WC)
2.0 Logistics
2.1 Army Enterprise Systems Integration Program (AESIP)
2.2 Global Combat Support System-Army (GCSS-Army)
2.3 Joint-Automatic Identification Technology (J-AIT)
2.4 Logistics Modernization Program (LMP)
2.5 Movement Tracking System (MTS)
2.6 Transportation Information Systems (TIS)
3.0 Human Capital
3.1 Army Human Resource System (AHRS)
3.2 Distributed Learning System (DLS)
3.3 Force Management System (FMS)
3.4 HR Solutions
3.5 Installation Management Systems-Army (IMS-A)
3.6 Integrated Personnel and Pay – Army (IPPS-A)
3.7 Reserve Component Automation Systems (RCAS)
4.0 Finance
4.1 General Fund Enterprise Business System (GFEBS)
5.0 Medical
5.1 Medical Communications for Combat Casualty Care (MC4)
6.0 Acquisition
6.1 Acquisition Business (AcqBusiness)
7.0 Biometrics
7.1 Department of Defense (DoD) Biometrics (DoD Biometrics)
7.2 Biometrics Enabling Capability (BEC)
7.3 Tactical Biometric Systems (TBS)

ALTESS

Probability of Program Success Operations Guide

Guidance to increase the PoPs

No matter what the outcome is this conference, we must seek ways to increase the Probability of Program Success not just apply new methods

The PoPS Operations Guide for ALTESS is shown highlighted here.

Starting at the top means asking a simple, yet powerful question, of any procurement processes. The two documents with larger borders are guidance from the IT initiatives. The other documents provide actionable outcomes for "increasing the probability of program success"

What is the probability of success?

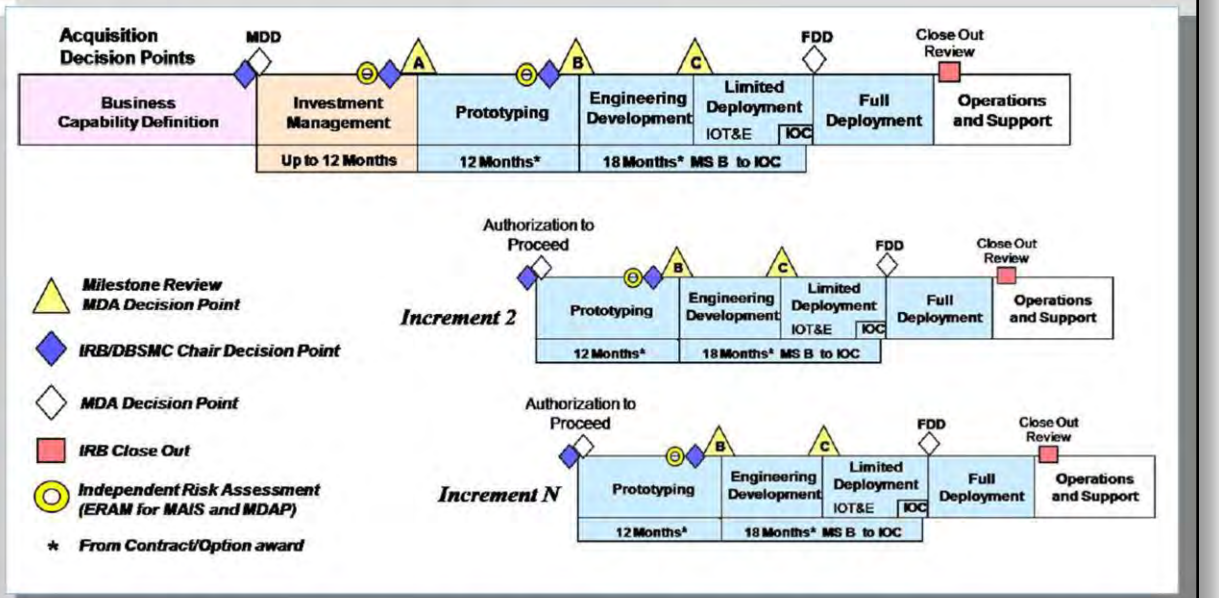
This is a legitimate question for any endeavor that evolves risk.

The processes and methods being described over the 3 days of this conference should be asking and answering the question:

- *how can we increase the probability of program success PoPS?*
- *How can we "connect the dots" between the proposed methods – agile methods – and the increase in PoPS?*
- *Same question needs to be asked of Earned Value, or for that matter any process – existing or proposed.*

A Quick View of the DoD IT Acquisition Lifecycle's Impact on Agile

5



Acquisition Decision Points

MDD

A

B

C

FDD

Close Out Review

Business Capability Definition

Investment Management

Prototyping

Engineering Development

Limited Deployment

IOT&E

IOC

Full Deployment

Operations and Support

Up to 12 Months

12 Months*

18 Months* MS B to IOC

Authorization to Proceed

Increment 2

Increment N

Milestone Review MDA Decision Point

IRB/DBSMC Chair Decision Point

MDA Decision Point

IRB Close Out

Independent Risk Assessment (ERAM for MAIS and MDAP)

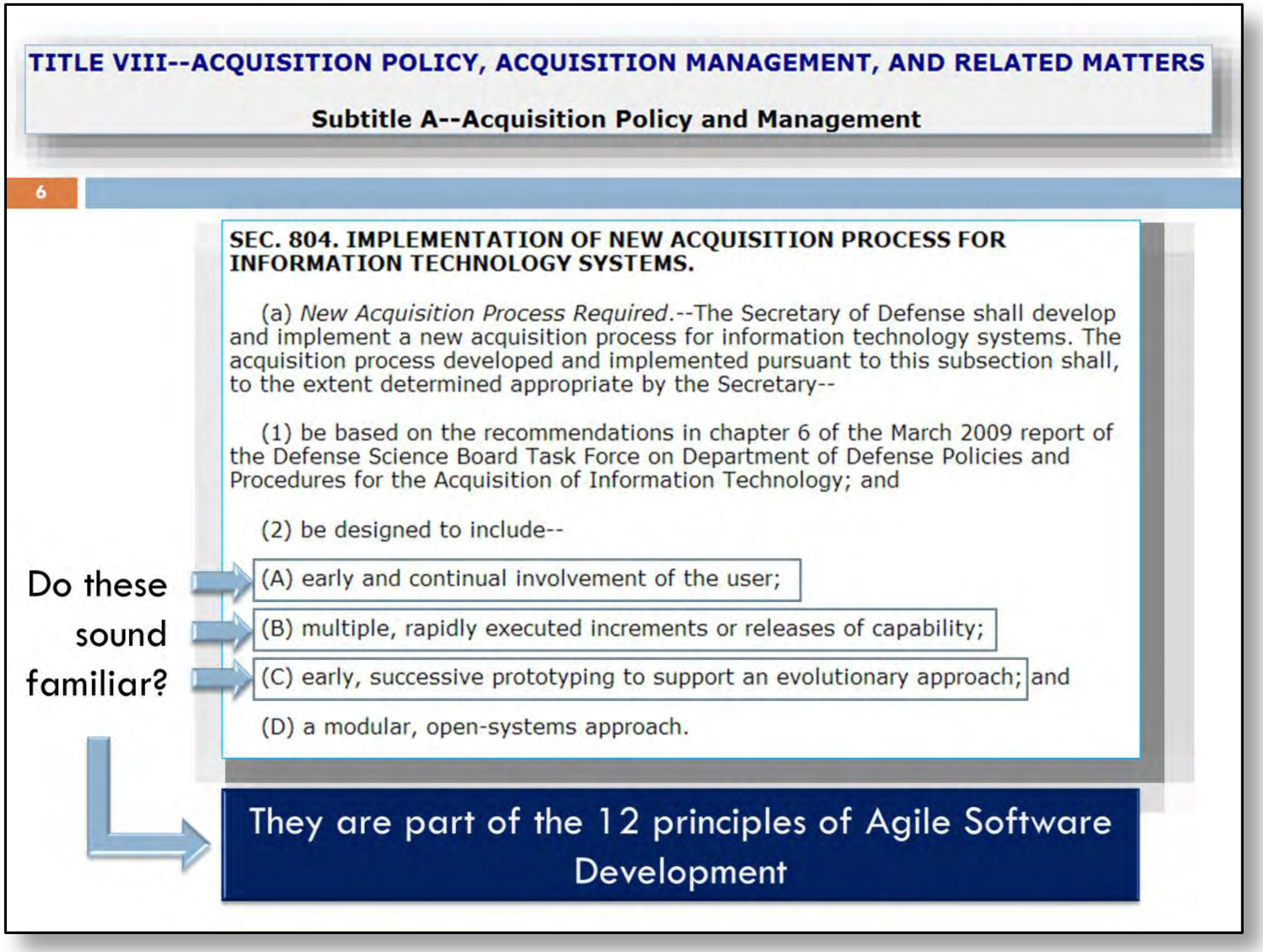
* From Contract/Option award

Earned Value requirement is stated in Table 3, Pg. 30, *A New Approach for Delivering Information Technology Capabilities in the Department of Defense*

Before we go any further, let's establish the connection between the need for agility in DoD IT procurement and Earned Value Management.

Page 30, Table 3 of *A New Approach for Delivering Information Technology Capabilities in the Department of Defense.*

this document, which you can find on the web, is from the Deputy Secretary of Defense, Office of the Deputy Chief Management Officer,



TITLE VIII--ACQUISITION POLICY, ACQUISITION MANAGEMENT, AND RELATED MATTERS

Subtitle A--Acquisition Policy and Management

6

SEC. 804. IMPLEMENTATION OF NEW ACQUISITION PROCESS FOR INFORMATION TECHNOLOGY SYSTEMS.

(a) *New Acquisition Process Required*.--The Secretary of Defense shall develop and implement a new acquisition process for information technology systems. The acquisition process developed and implemented pursuant to this subsection shall, to the extent determined appropriate by the Secretary--

(1) be based on the recommendations in chapter 6 of the March 2009 report of the Defense Science Board Task Force on Department of Defense Policies and Procedures for the Acquisition of Information Technology; and

(2) be designed to include--

(A) early and continual involvement of the user;

(B) multiple, rapidly executed increments or releases of capability;

(C) early, successive prototyping to support an evolutionary approach; and

(D) a modular, open-systems approach.

Do these sound familiar?

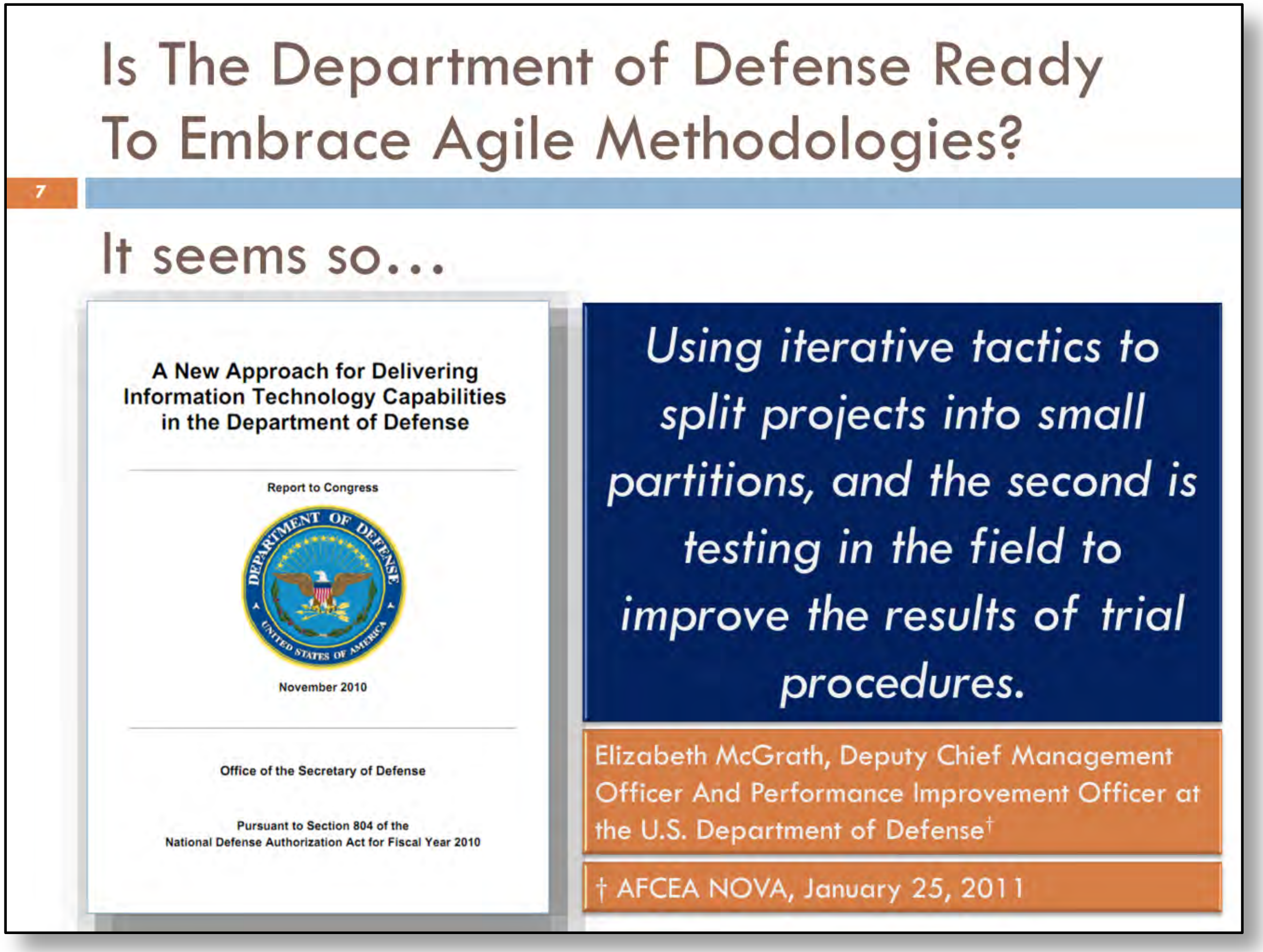
They are part of the 12 principles of Agile Software Development

So if we're looking for a higher motivation in our search for corrective actions to being over budget and behind schedule, we need look no further than the current NDAA.

Here's the actual worlds from the NDAA. If you have not read this, it would worthwhile. The NDAA is interesting in that it is a "directive" from SecDef to the DoD IT community.

It provides clear and concise statements about what to search for. A, B, and C say it in clear terms.

- Early and continuous user involvement
- Rapidly executed increments or released of capability. Capability is a DoD term (Capability Based Planning is a DoD process). Capability means "I can do something with the thing you just gave me."
- Early successive prototyping to support an evolutionary approach – means what it says. Early – not late, evolutionary – not big bang, prototyping – partially complete things that can be examined to see if that's what we really want.



Is The Department of Defense Ready To Embrace Agile Methodologies?

7

It seems so…

A New Approach for Delivering Information Technology Capabilities in the Department of Defense

Report to Congress

November 2010

Office of the Secretary of Defense

Pursuant to Section 804 of the National Defense Authorization Act for Fiscal Year 2010

Using iterative tactics to split projects into small partitions, and the second is testing in the field to improve the results of trial procedures.

Elizabeth McGrath, Deputy Chief Management Officer And Performance Improvement Officer at the U.S. Department of Defense†

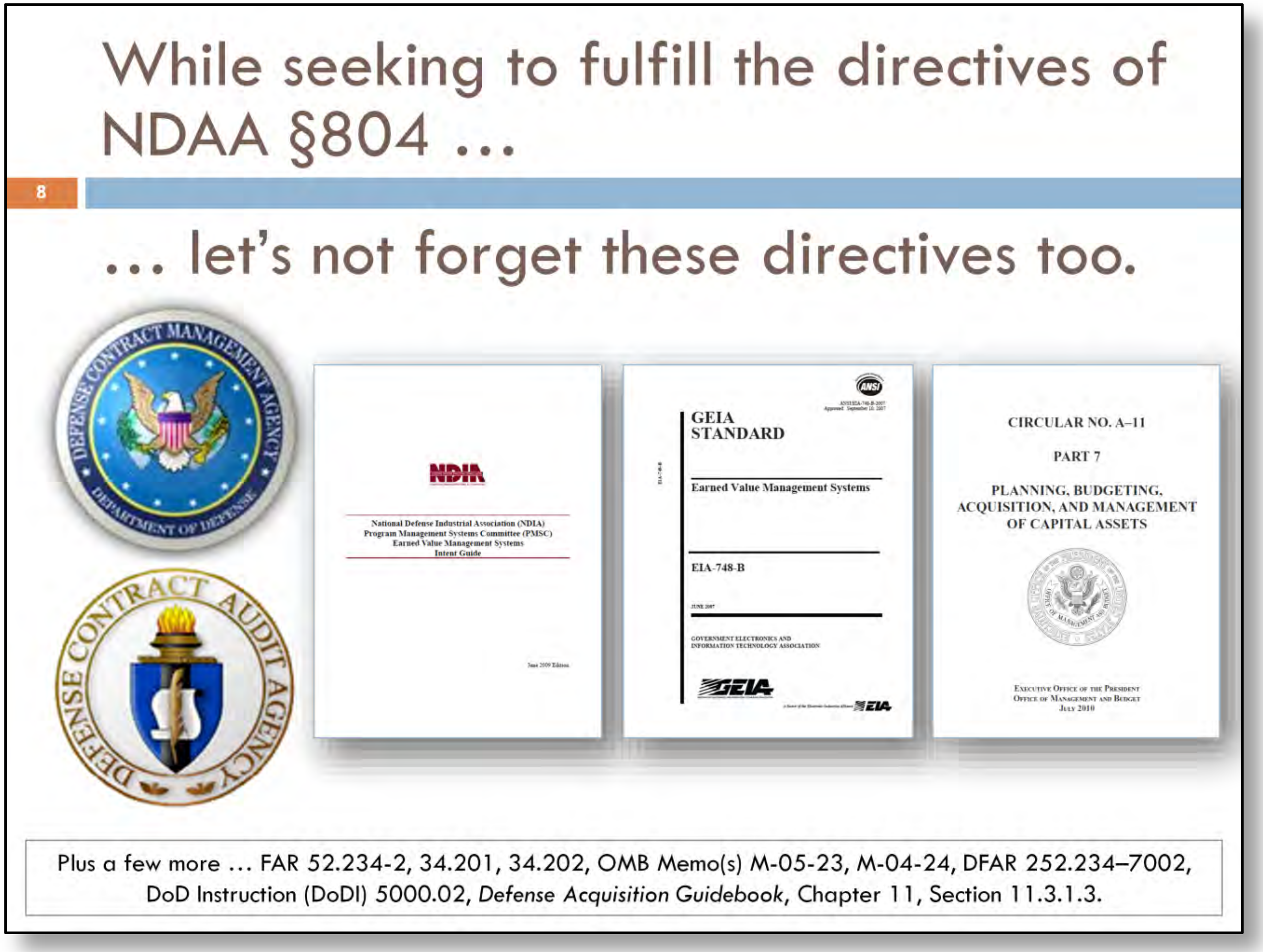
† AFCEA NOVA, January 25, 2011

In the presence of all these myths – procurement, DoD IT, and Agile Software Development, here is ample evidence DoD IT is headed down the path of agile acquisition and development.

Mrs. McGrath spoke at a recent AFCEA NOVA lunch I attended and laid out where she was going in her office.

But we still need to “connect the dots” between the Governance of DoD IT programs and the technical activities we find in the development of software. As mentioned earlier “writing software” is not the same as “managing the writing of software.”

No matter the examples in the commercial worlds, where the development teams are “self managed,” that is likely too big a leap for FAR / DFAR compliant programs to take. There will always be the requirement for Program Management processes based on Earned Value for contract awards greater than $20M.



While seeking to fulfill the directives of NDAA §804 …

… let's not forget these directives too.

8
DEFENSE CONTRACT MANAGEMENT AGENCY
DEPARTMENT OF DEFENSE
DEFENSE CONTRACT AUDIT AGENCY
NDIA
National Defense Industrial Association (NDIA)
Program Management Systems Committee (PMSC)
Earned Value Management Systems
Intent Guide
GEIA STANDARD
Earned Value Management Systems
EIA-748-B
GOVERNMENT ELECTRONICS AND INFORMATION TECHNOLOGY ASSOCIATION
GEIA
EIA
CIRCULAR NO. A–11
PART 7
PLANNING, BUDGETING, ACQUISITION, AND MANAGEMENT OF CAPITAL ASSETS
EXECUTIVE OFFICE OF THE PRESIDENT
OFFICE OF MANAGEMENT AND BUDGET
JULY 2010

Plus a few more … FAR 52.234-2, 34.201, 34.202, OMB Memo(s) M-05-23, M-04-24, DFAR 252.234–7002, DoD Instruction (DoDI) 5000.02, *Defense Acquisition Guidebook*, Chapter 11, Section 11.3.1.3.

So now that we've had a good tour of agile some myths busted or confirmed, and the interaction of agile with the project and the development of software, let's revisit that some guidance that is in place no matter what software development we're using now or want to use in the future.

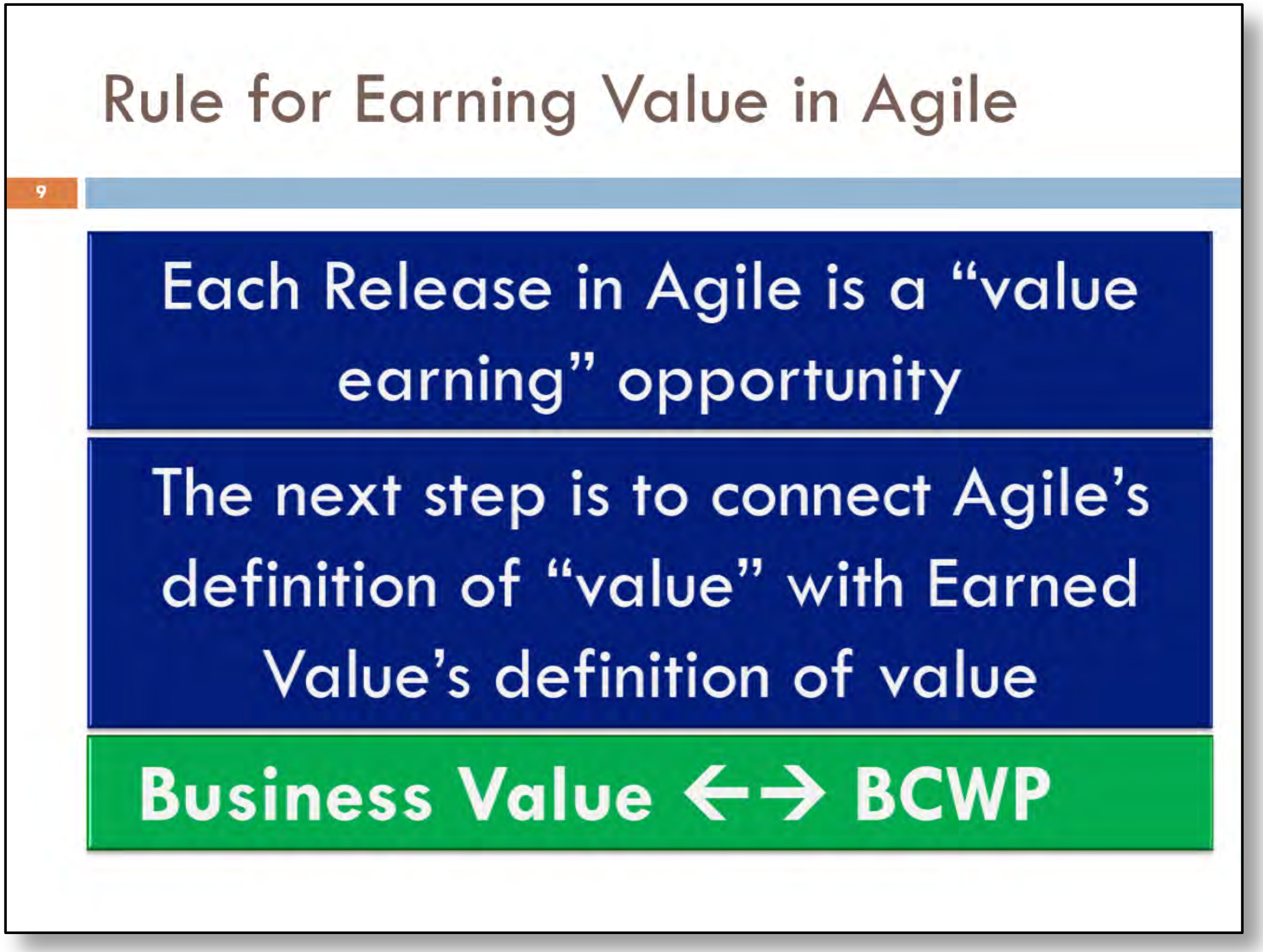
We come to the elephant in the room.

For programs in the DoD (or for that matter any government agency) that have award values greater than $20M the FAR, DFAR, and OMB (White House) requires Earned Value management, guided by ANSI 748-B.

I'll wait for the shudder in the room to settle (if there is one).

The two logos on the left are from the Defense Contract Management Agency and the Defense Contract Audit Agency. They are accountable for looking after the money issued to contractors for the acquisition of services and materials in the US Government. They are one of those overworked agencies that are always looking for ways to make your life unpleasant at inconvenient times.

They do this with a "politically correct word" surveillance – which mean audit – enabled by the regulations and guidance listed at the bottom of this chart.



Rule for Earning Value in Agile

9

Each Release in Agile is a “value earning” opportunity

The next step is to connect Agile’s definition of “value” with Earned Value’s definition of value

Business Value ←→ BCWP

Let’s bring the discussion back to some simple, clear, and concise terms.

What are we after when I suggest Earned Value Management can be used with Agile Development?

Actually in the Federal procurement domain, it’s agile being used with Earned Value.

The answer is “how can we recognize that value – business value – is being EARNED in exchange for spending time and money?”

This is a core question, in the same way to previous question – what is the probability of program success – is a core question.

If we proceed further without understand the importance of these core questions, we have heard and seen some very cleaver tools and approaches. But we won’t understand WHY they are cleaver. And most importantly if they are in fact the appropriate approaches to the problem.

And we all understand the problem right?

We’re over budget, behind schedule, and off the technical performance measures on many programs in IT and other DoD procurement domains.

Before Going Further, Let's Bust Some Myths

10

So let's change course here for a bit.

There are lots of "myths" around agile software development. Just like there are lots of myths around Earned Value and Earned Value Management.

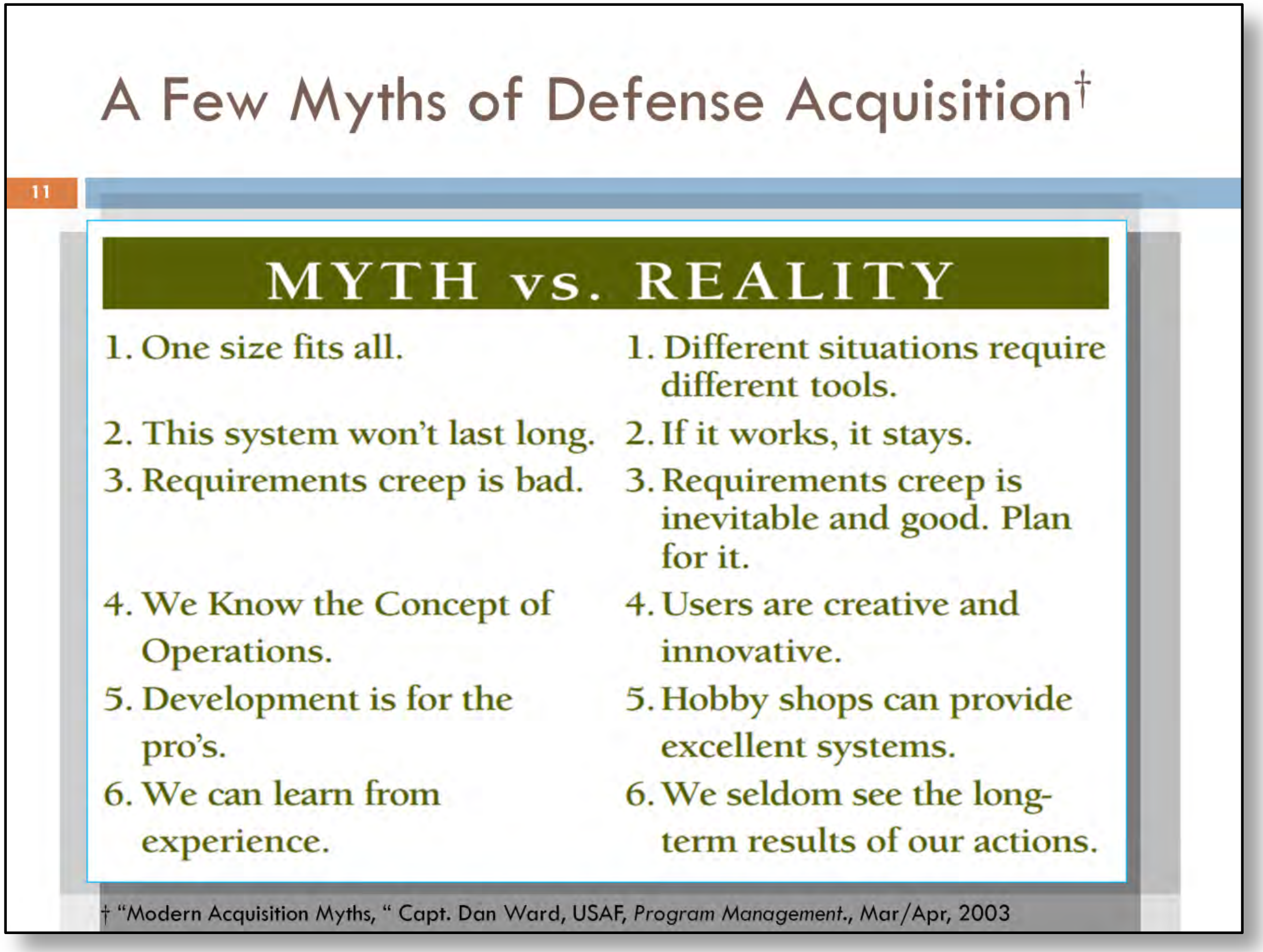
Let's look at some of these to get a sense if these myths have any validity to them.

If not let's bust them.

If so, let's use them to make improvements in our understanding of what to do next to Increase the Probability of Program Success.

Remember that phrase. That's the phrase we want to start using to keep everyone honest.

How does your suggested improvement Increase the Probability of Program Success?



A Few Myths of Defense Acquisition†

MYTH vs. REALITY	
1. One size fits all.	1. Different situations require different tools.
2. This system won't last long.	2. If it works, it stays.
3. Requirements creep is bad.	3. Requirements creep is inevitable and good. Plan for it.
4. We Know the Concept of Operations.	4. Users are creative and innovative.
5. Development is for the pro's.	5. Hobby shops can provide excellent systems.
6. We can learn from experience.	6. We seldom see the long-term results of our actions.

† "Modern Acquisition Myths, " Capt. Dan Ward, USAF, *Program Management.*, Mar/Apr, 2003

Let's start with some myths no the Defense Acquisition side.

These come from then Capt. Dan Ward, now Lt. Col Dan Ward, USAF.

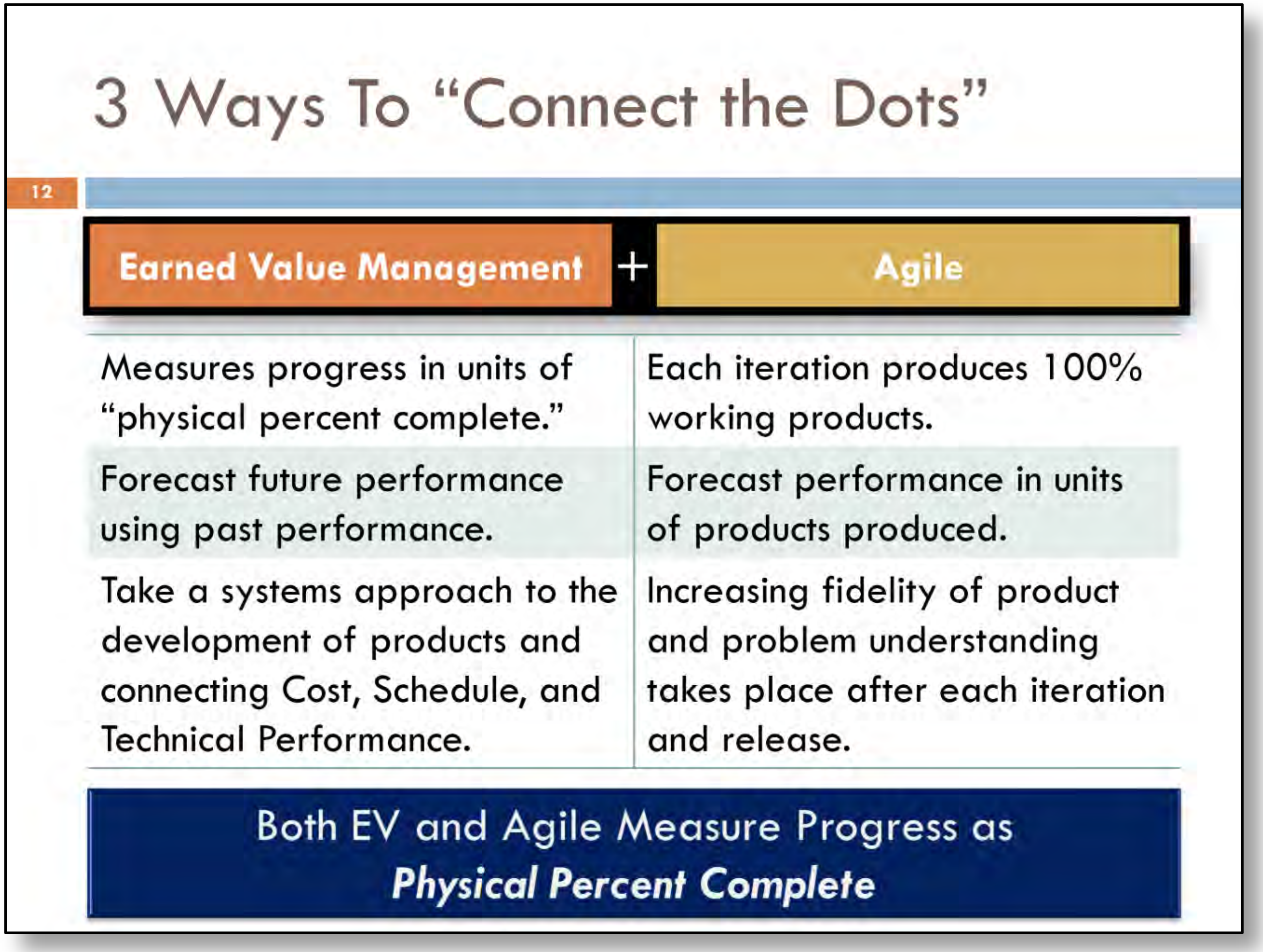
Dan and I have shared ideas for awhile around what it means to be agile and adaptive in the weapons system procurement business.

Dan writes articles for the Acquisition, Technology and Logistics journal – a real page turner if anyone is interested.

Dan also has a Blog and writes books about management, especially program management.

Most of Dan's work can be found on the Defense Acquisition University's Community of Practice portal.

These myths are self evident. Meaning when you statement them, you can figure pretty quickly if they can be "busted" or not. There are 6 here, all "busted."



3 Ways To "Connect the Dots"

12

Earned Value Management +	Agile
Measures progress in units of "physical percent complete."	Each iteration produces 100% working products.
Forecast future performance using past performance.	Forecast performance in units of products produced.
Take a systems approach to the development of products and connecting Cost, Schedule, and Technical Performance.	Increasing fidelity of product and problem understanding takes place after each iteration and release.

Both EV and Agile Measure Progress as ***Physical Percent Complete***

We're getting close to the half way point in this briefing, so let's have a process check.

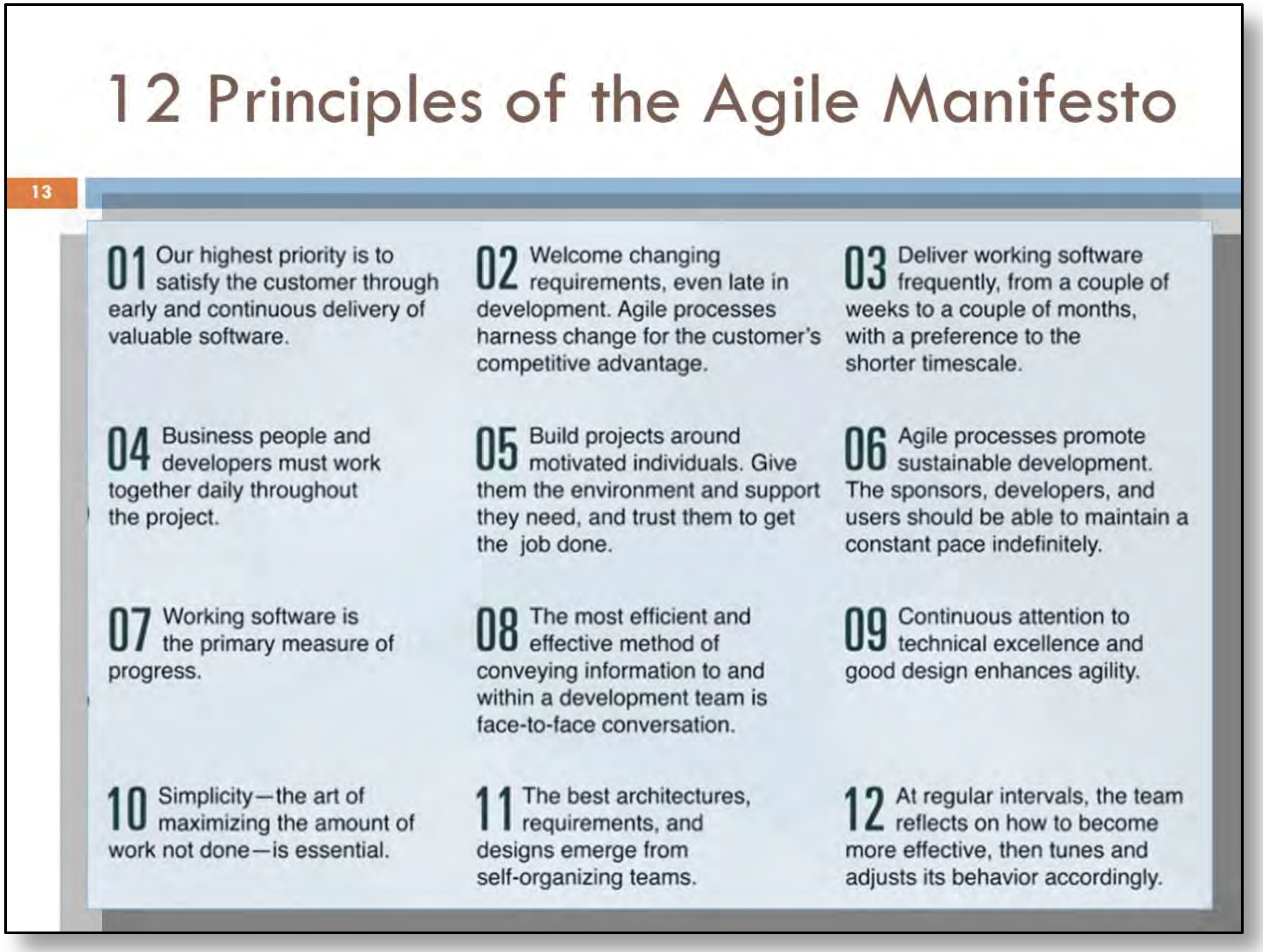
First where have we come from? We've seen agile is being mentioned inside the walls of the DoD.

We've seen there are external guiding regulations and documents that impact DoD procurement no matter what method is being used to develop the software.

So let's take the first attempt to "connect the dots," between those two worlds.

Here's three ways they can be connected.

- Measuring progress
- Forecasting future progress
- Integrating the performance reporting in a form needed by the government.



12 Principles of the Agile Manifesto

13

01 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

02 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

03 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

04 Business people and developers must work together daily throughout the project.

05 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

06 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

07 Working software is the primary measure of progress.

08 The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

09 Continuous attention to technical excellence and good design enhances agility.

10 Simplicity—the art of maximizing the amount of work not done—is essential.

11 The best architectures, requirements, and designs emerge from self-organizing teams.

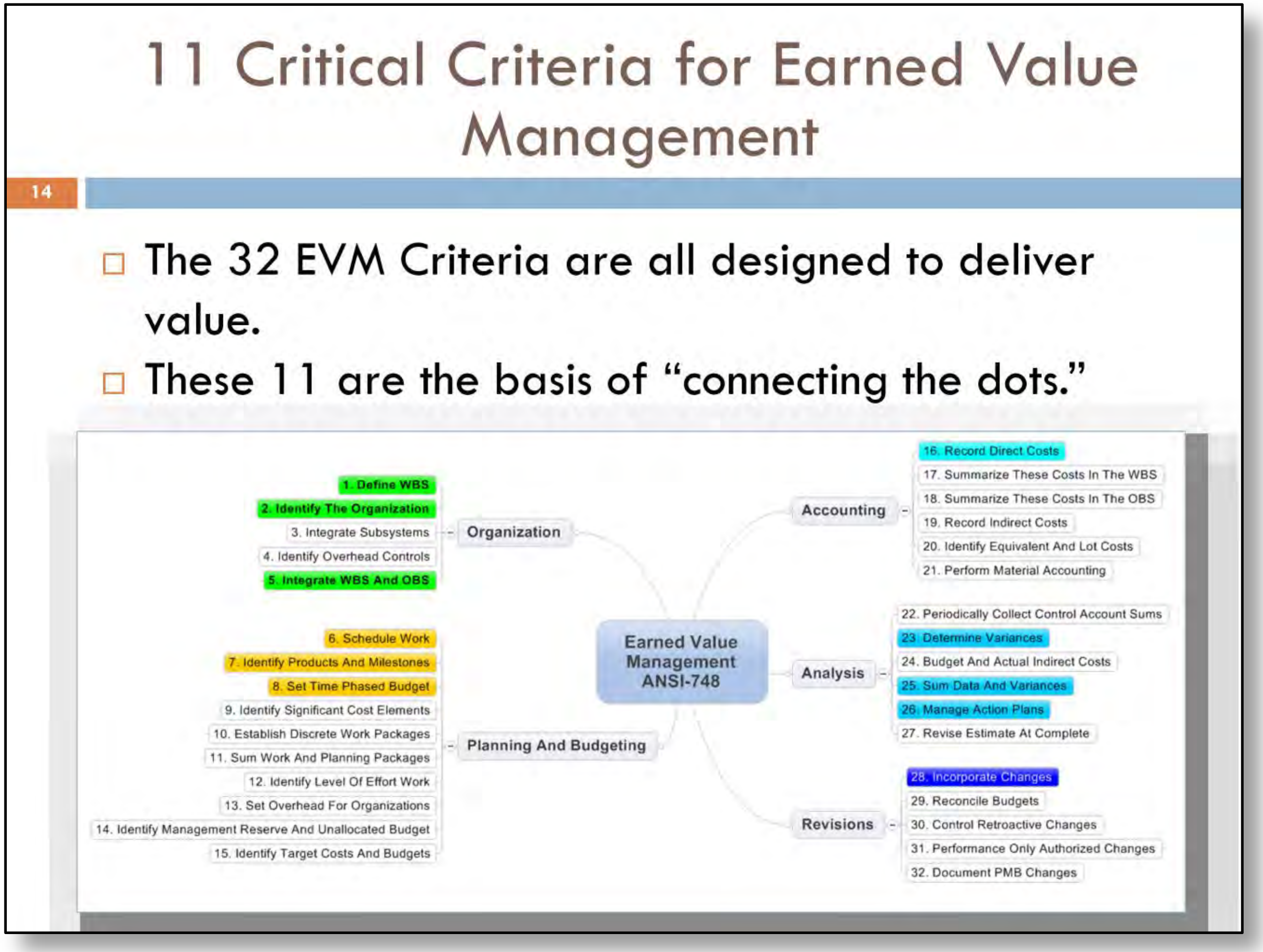
12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

One of the difficulties with the Agile Manifesto besides the term “over,” is it is not directly actionable.

If we look at these 12 “principles” and remove the term “agile” there is not one of them that we would not want on any project.

How would not want…

- To satisfy the customer with early and continuous delivery of value
- To have business and developers work together.
- To frequently deliver working products.
- To have continuous attention to technical excellence.



11 Critical Criteria for Earned Value Management

14

- □ The 32 EVM Criteria are all designed to deliver value.
- □ These 11 are the basis of "connecting the dots."

Earned Value Management ANSI-748

Organization
- 1. Define WBS
- 2. Identify The Organization
- 3. Integrate Subsystems
- 4. Identify Overhead Controls
- 5. Integrate WBS And OBS

Planning And Budgeting
- 6. Schedule Work
- 7. Identify Products And Milestones
- 8. Set Time Phased Budget
- 9. Identify Significant Cost Elements
- 10. Establish Discrete Work Packages
- 11. Sum Work And Planning Packages
- 12. Identify Level Of Effort Work
- 13. Set Overhead For Organizations
- 14. Identify Management Reserve And Unallocated Budget
- 15. Identify Target Costs And Budgets

Accounting
- 16. Record Direct Costs
- 17. Summarize These Costs In The WBS
- 18. Summarize These Costs In The OBS
- 19. Record Indirect Costs
- 20. Identify Equivalent And Lot Costs
- 21. Perform Material Accounting

Analysis
- 22. Periodically Collect Control Account Sums
- 23. Determine Variances
- 24. Budget And Actual Indirect Costs
- 25. Sum Data And Variances
- 26. Manage Action Plans
- 27. Revise Estimate At Complete

Revisions
- 28. Incorporate Changes
- 29. Reconcile Budgets
- 30. Control Retroactive Changes
- 31. Performance Only Authorized Changes
- 32. Document PMB Changes

ANSI-748-B defines 32 criteria needs for a FAR/DFAR compliant Earned Value Management System.

These criteria address 5 areas of Earned Value Management

1. Organization
2. Planning and Budgeting
3. Accounting
4. Analysis
5. Revisions

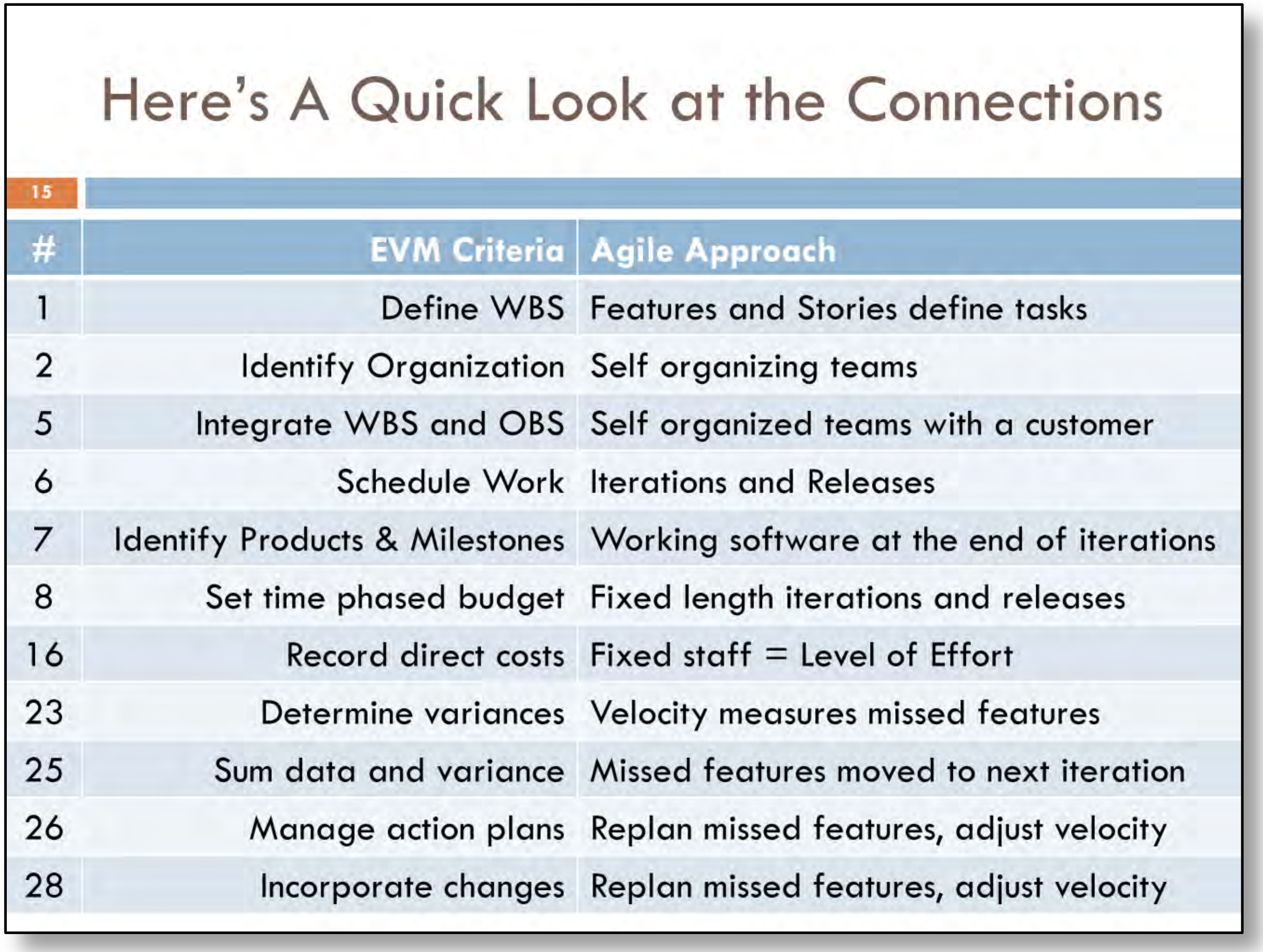
These areas are the 5 critical success factors for any program whether it is managed with Earned Value or not and whether Agile Software development methods are used or not.

These 5 program management processes are the basis of Increasing the Probability of Success of any program.

But there are 11 critical criteria that must be present not matter what approach is taken to the management of a program.

They ask and answer questions that provider actionable information to the Program Manager.

It is these 11 critical criteria that we'll connect with the principles of Agile Software Development.



Here’s A Quick Look at the Connections

15

#	EVM Criteria	Agile Approach
1	Define WBS	Features and Stories define tasks
2	Identify Organization	Self organizing teams
5	Integrate WBS and OBS	Self organized teams with a customer
6	Schedule Work	Iterations and Releases
7	Identify Products & Milestones	Working software at the end of iterations
8	Set time phased budget	Fixed length iterations and releases
16	Record direct costs	Fixed staff = Level of Effort
23	Determine variances	Velocity measures missed features
25	Sum data and variance	Missed features moved to next iteration
26	Manage action plans	Replan missed features, adjust velocity
28	Incorporate changes	Replan missed features, adjust velocity

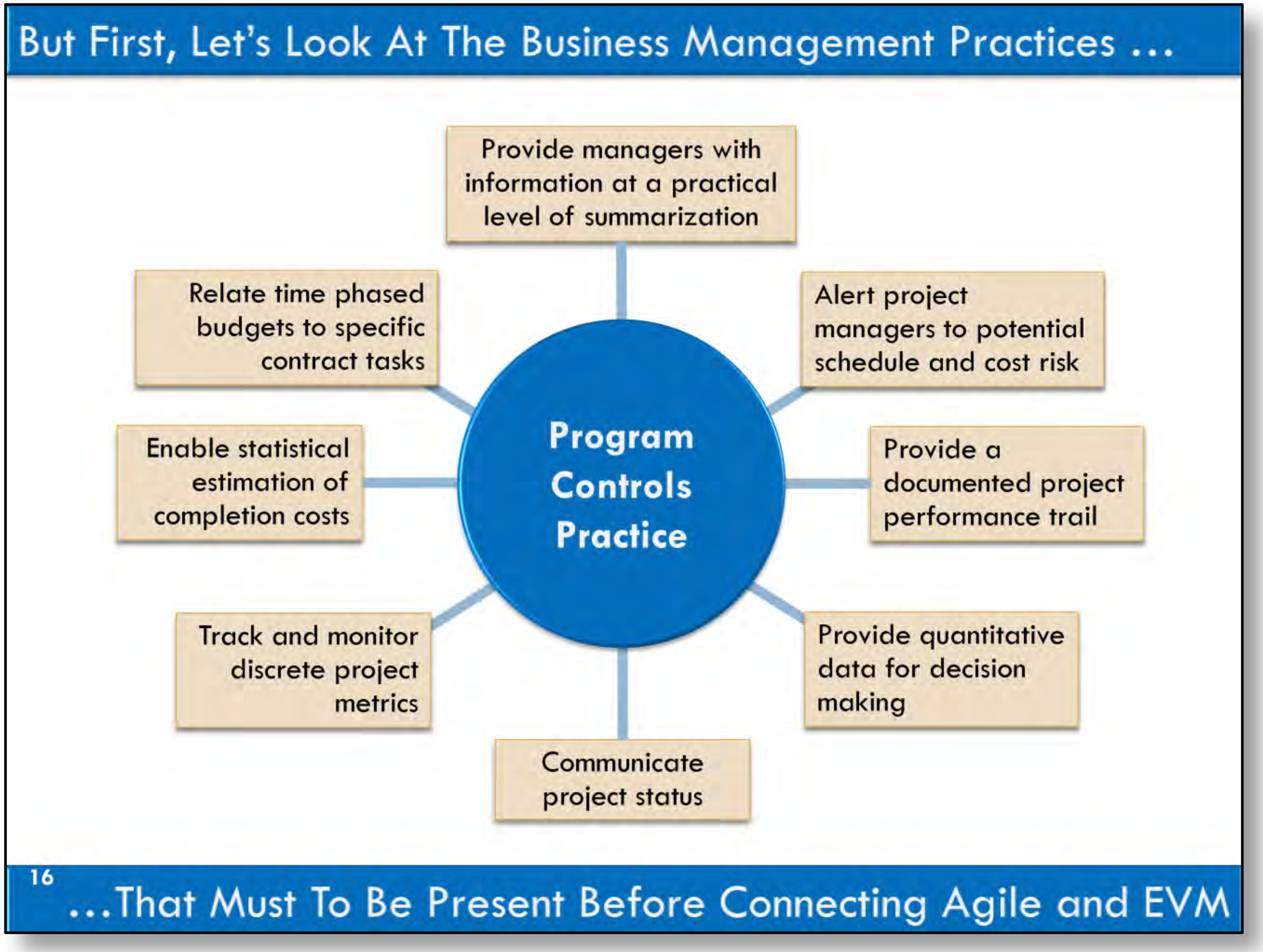
Describing how to make these connections and deploying them on actual programs is beyond the scope of this brief introduction.

But here is a quick look at how the connections are related.

In Agile Software Development the 12 principles that we saw previously fit nicely with the 11 Earned Value Management criteria.

Both Earned Value and Agile Software Development share several important principles:

1. Progress is measured through physical measures of complete.
2. Planning is incremental and iterative.
3. Measures of Effectiveness and Measures of Performance are developed through customer interaction.
4. Work is organized to produce tangible outcomes.
5. Changes are managed with the full involvement of the customer.
6. Adjusts to forecasted performance are made from measures of past performance.



But First, Let's Look At The Business Management Practices …

Program Controls Practice

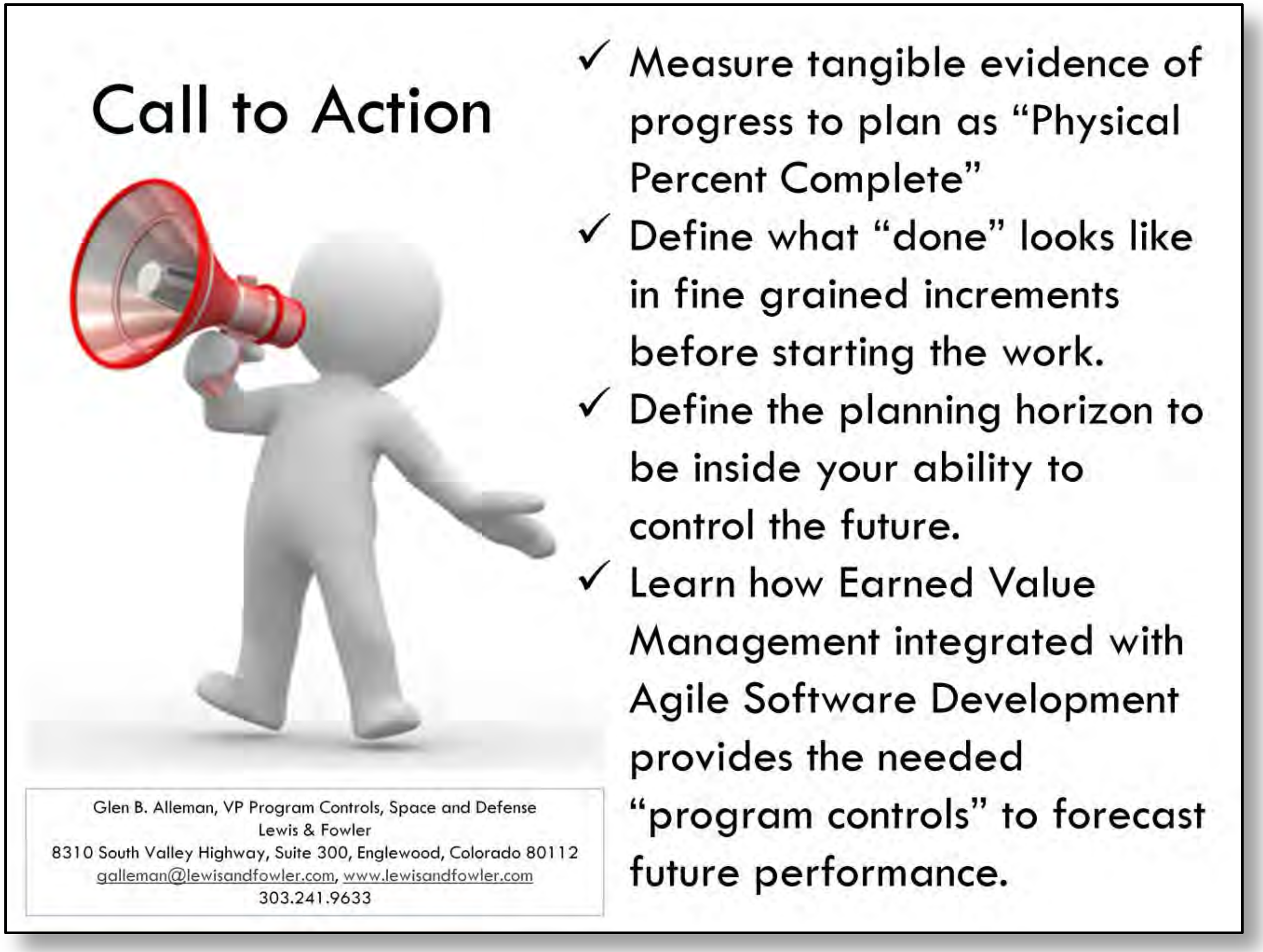
- Provide managers with information at a practical level of summarization
- Alert project managers to potential schedule and cost risk
- Provide a documented project performance trail
- Provide quantitative data for decision making
- Communicate project status
- Track and monitor discrete project metrics
- Enable statistical estimation of completion costs
- Relate time phased budgets to specific contract tasks

…That Must To Be Present Before Connecting Agile and EVM

No matter how we connect the dots between Earned Value Management and Agile Software Development, there is principles of business management that must be in place. These principles must drive the deployment of both Agile Software Development and Earned Value Management.

They are obvious when arranged in this way. No credible IT manager would object to the application of these principles.

So no matter how we proceed with the integration of Agile Development on DoD IT programs, processes should be in place that provide this information to the decision makers.



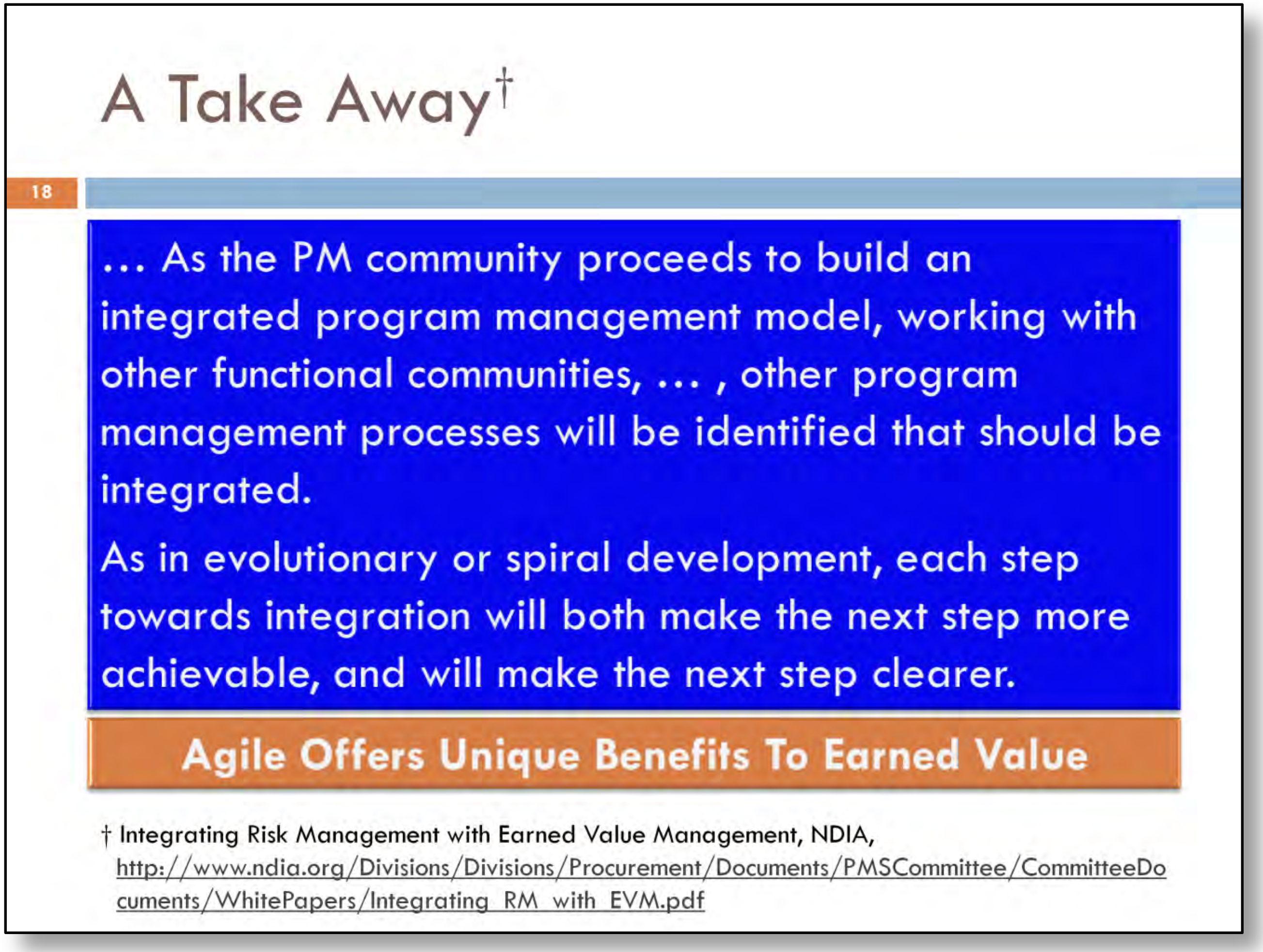
Call to Action

- ✓ Measure tangible evidence of progress to plan as "Physical Percent Complete"
- ✓ Define what "done" looks like in fine grained increments before starting the work.
- ✓ Define the planning horizon to be inside your ability to control the future.
- ✓ Learn how Earned Value Management integrated with Agile Software Development provides the needed "program controls" to forecast future performance.

Glen B. Alleman, VP Program Controls, Space and Defense
Lewis & Fowler
8310 South Valley Highway, Suite 300, Englewood, Colorado 80112
galleman@lewisandfowler.com, www.lewisandfowler.com
303.241.9633

How a suggested approach to moving forward with the integration of Agile Software Development in the domain of DoD IT Acquisition.

1. Assure all performance measurement baselines measure progress as "physical percent complete" in units of measure meaningful to the decision makers.
2. Define what "done" looks like on fine grained boundaries with tangible evidence, agreed to before starting the work.
3. Use Rolling Waves to bound the planning horizon inside our ability to control the future.
4. Integrate Agile Software Development into the DoD Program Controls paradigm it increase the visibility of performance to the decision makers.



A Take Away[†]

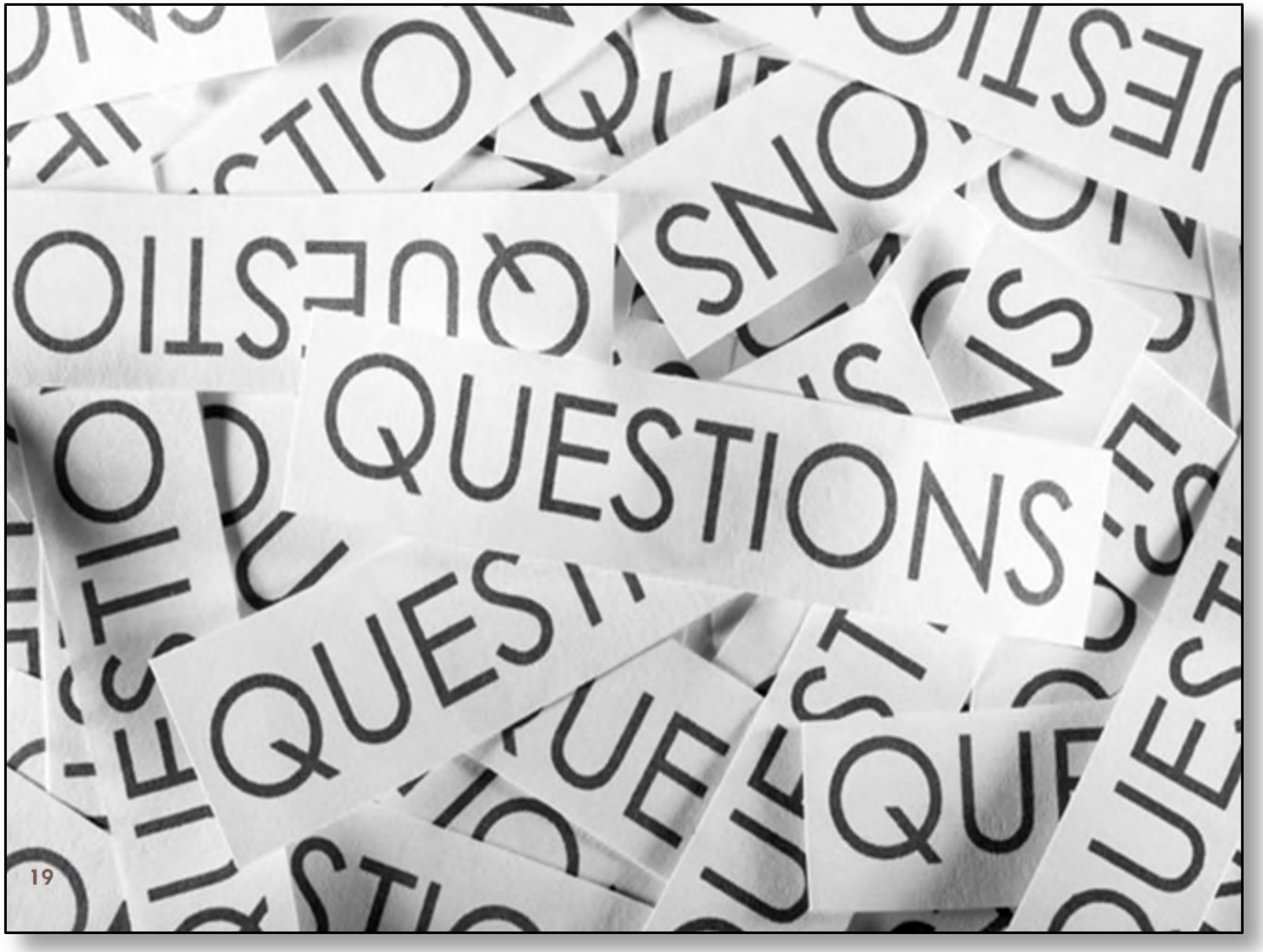
… As the PM community proceeds to build an integrated program management model, working with other functional communities, … , other program management processes will be identified that should be integrated.

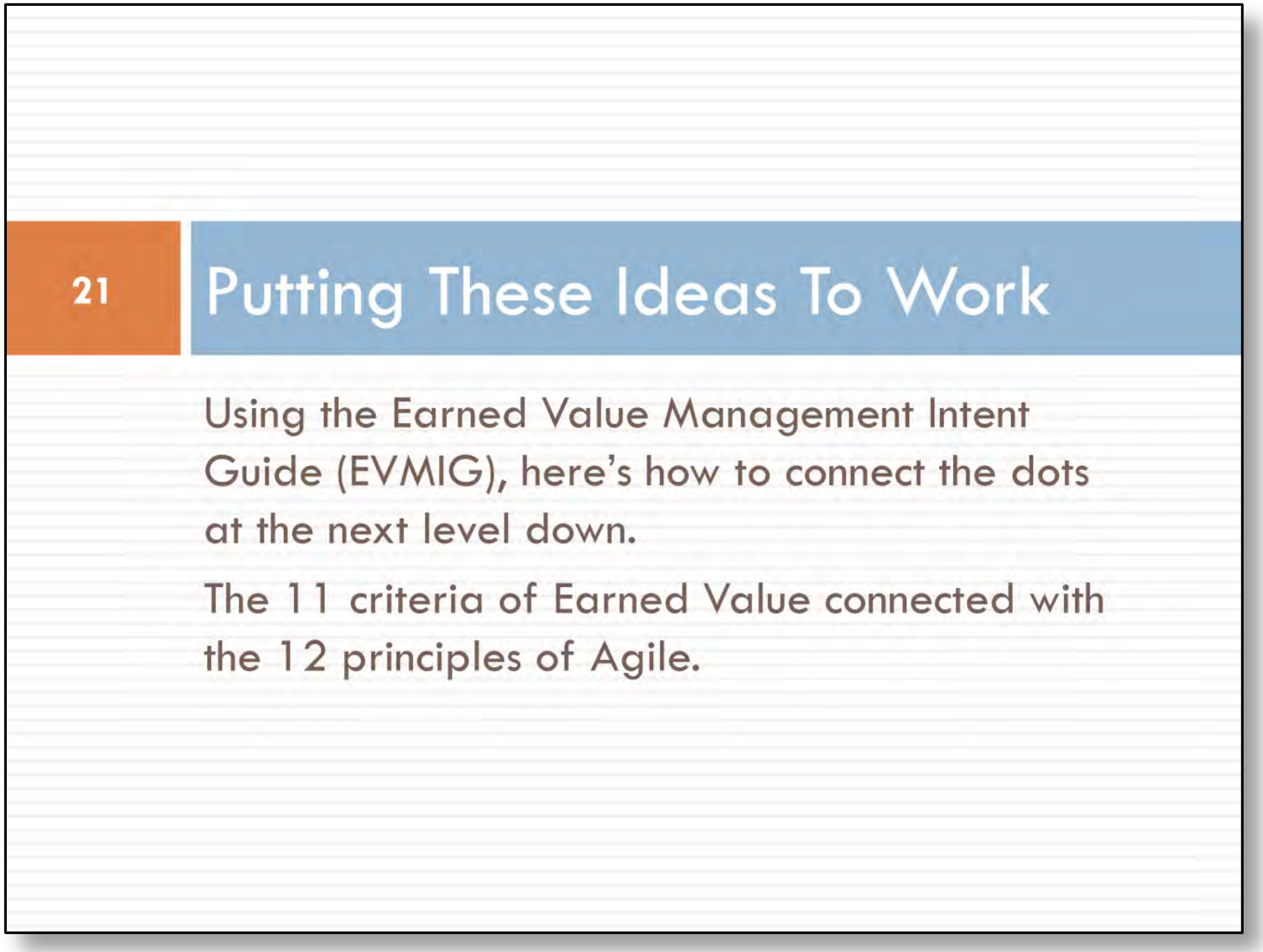
As in evolutionary or spiral development, each step towards integration will both make the next step more achievable, and will make the next step clearer.

Agile Offers Unique Benefits To Earned Value

[†] Integrating Risk Management with Earned Value Management, NDIA, http://www.ndia.org/Divisions/Divisions/Procurement/Documents/PMSCommittee/CommitteeDocuments/WhitePapers/Integrating_RM_with_EVM.pdf

Earned Value + Agile = Success
Glen B. Alleman, VP Program Controls, Lewis & Fowler
NDIA Information Systems Summit II
4/4/2011 – 4/6/2011
Hyatt Regency, Baltimore, Maryland





21

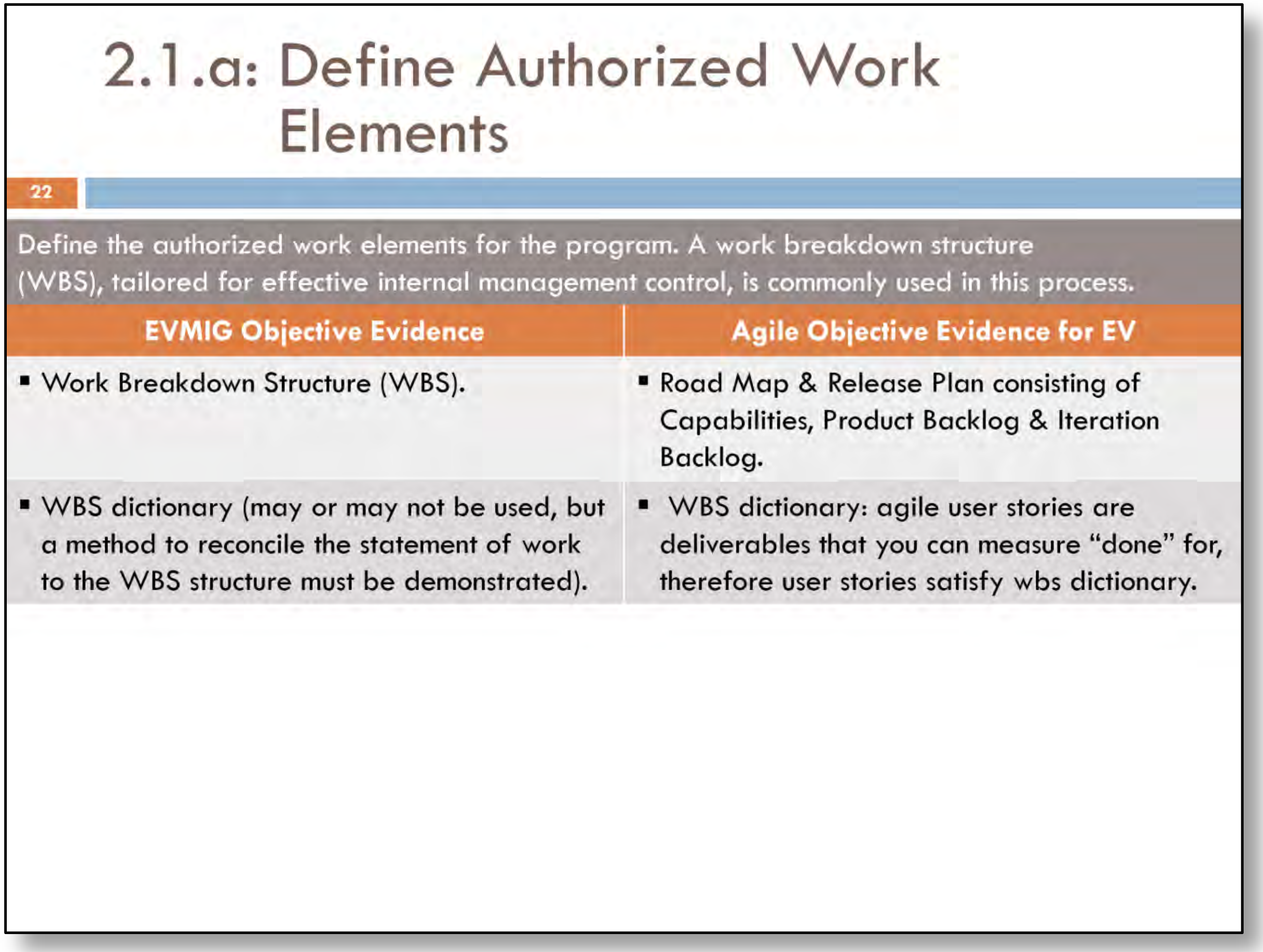
Putting These Ideas To Work

Using the Earned Value Management Intent Guide (EVMIG), here's how to connect the dots at the next level down.

The 11 criteria of Earned Value connected with the 12 principles of Agile.

With the 11 748-B Criteria, let's go down one more level and see how Agile Software development practices can be connected, using the NDIA Earned Value Management Intent Guide (EVMIG).

The numbers in the title section of the following pages are from the EVMIG.



2.1.a: Define Authorized Work Elements

22

Define the authorized work elements for the program. A work breakdown structure (WBS), tailored for effective internal management control, is commonly used in this process.

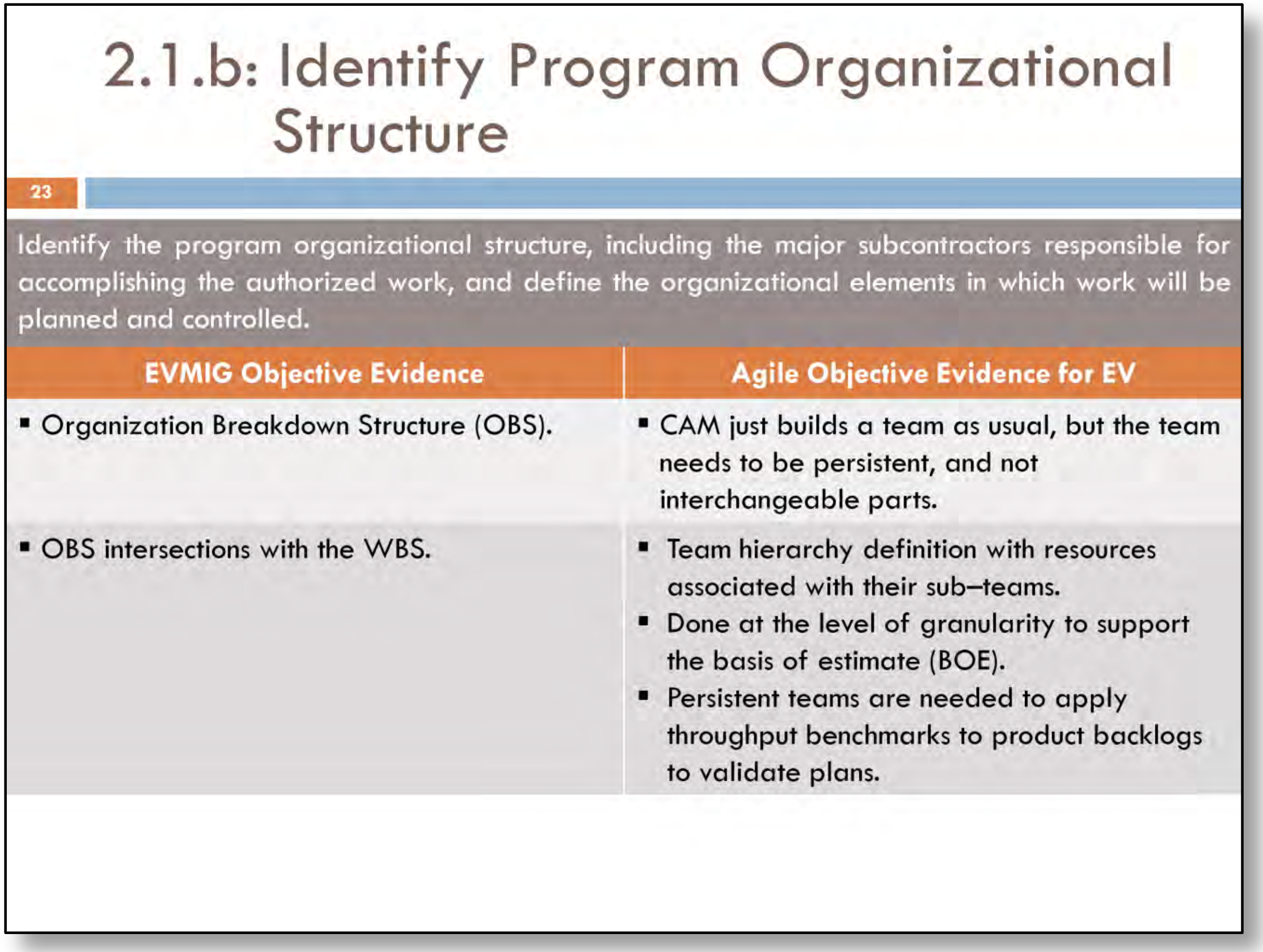
EVMIG Objective Evidence	Agile Objective Evidence for EV
▪ Work Breakdown Structure (WBS).	▪ Road Map & Release Plan consisting of Capabilities, Product Backlog & Iteration Backlog.
▪ WBS dictionary (may or may not be used, but a method to reconcile the statement of work to the WBS structure must be demonstrated).	▪ WBS dictionary: agile user stories are deliverables that you can measure "done" for, therefore user stories satisfy wbs dictionary.

2.1.a describes how to define what work is to be performed on the project. In the Agile paradigm this work might be considered "emerging." But during an iteration and during a release planning session, the defined work should be clear.

The same is true for a Rolling Wave planning process.

In 748-B we need a Work Breakdown Structure. For EV programs this is defined in MIL-STD-881C (coming out in June of 2011). For Agile, the release planning process produces artifacts that describe what is to be produced. These can be "sticky" notes all the way up to reports from supporting tools.

The WBS Dictionary is a narrative of what is to be delivered during the work efforts. Agile provides "stories" or other narrative forms that perform the same function.



2.1.b: Identify Program Organizational Structure

23

Identify the program organizational structure, including the major subcontractors responsible for accomplishing the authorized work, and define the organizational elements in which work will be planned and controlled.

EVMIG Objective Evidence	Agile Objective Evidence for EV
▪ Organization Breakdown Structure (OBS).	▪ CAM just builds a team as usual, but the team needs to be persistent, and not interchangeable parts.
▪ OBS intersections with the WBS.	▪ Team hierarchy definition with resources associated with their sub–teams. ▪ Done at the level of granularity to support the basis of estimate (BOE). ▪ Persistent teams are needed to apply throughput benchmarks to product backlogs to validate plans.

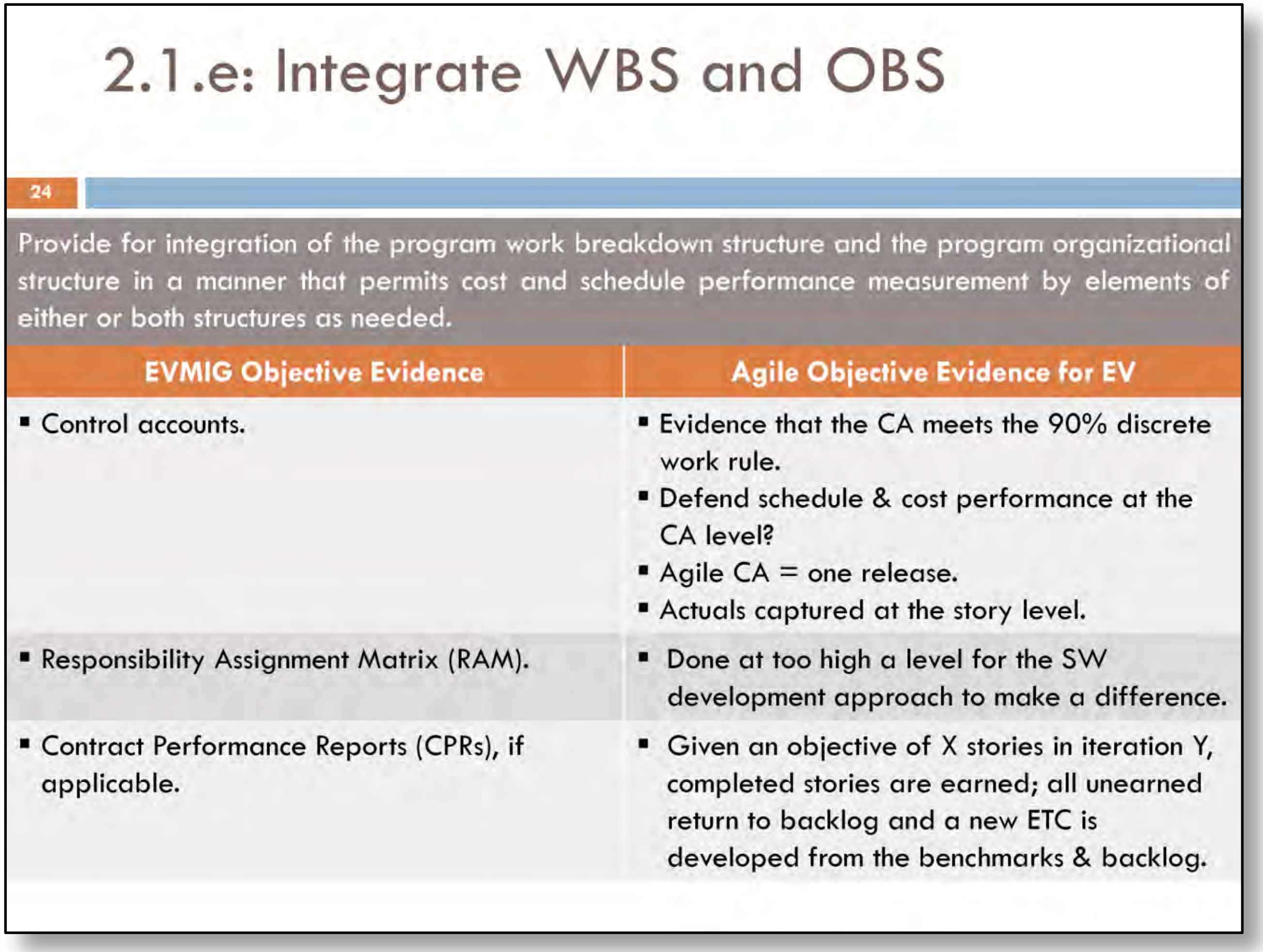
2.1.b defines who is working on the project. In the Earned Value world this issue is more complex, because people come and go on the project. The Organizational Breakdown Structure provides this information.

On Agile projects, the staff is fixed for the most part during the iteration and possibly across the release cycle.

The artifacts in Agile that describe the staff can be simple and be posted in the wall.

The motivation for the OBS and the WBS in 2.1.a in Earned Value is to define the Control Account and the Control Account Manager (CAM) responsible for the delivery of the items in the Control Account.

Depending on the size of the project, similar formal processes will be needed for Agile.



2.1.e: Integrate WBS and OBS

24

Provide for integration of the program work breakdown structure and the program organizational structure in a manner that permits cost and schedule performance measurement by elements of either or both structures as needed.

EVMIG Objective Evidence	Agile Objective Evidence for EV
▪ Control accounts.	▪ Evidence that the CA meets the 90% discrete work rule. ▪ Defend schedule & cost performance at the CA level? ▪ Agile CA = one release. ▪ Actuals captured at the story level.
▪ Responsibility Assignment Matrix (RAM).	▪ Done at too high a level for the SW development approach to make a difference.
▪ Contract Performance Reports (CPRs), if applicable.	▪ Given an objective of X stories in iteration Y, completed stories are earned; all unearned return to backlog and a new ETC is developed from the benchmarks & backlog.

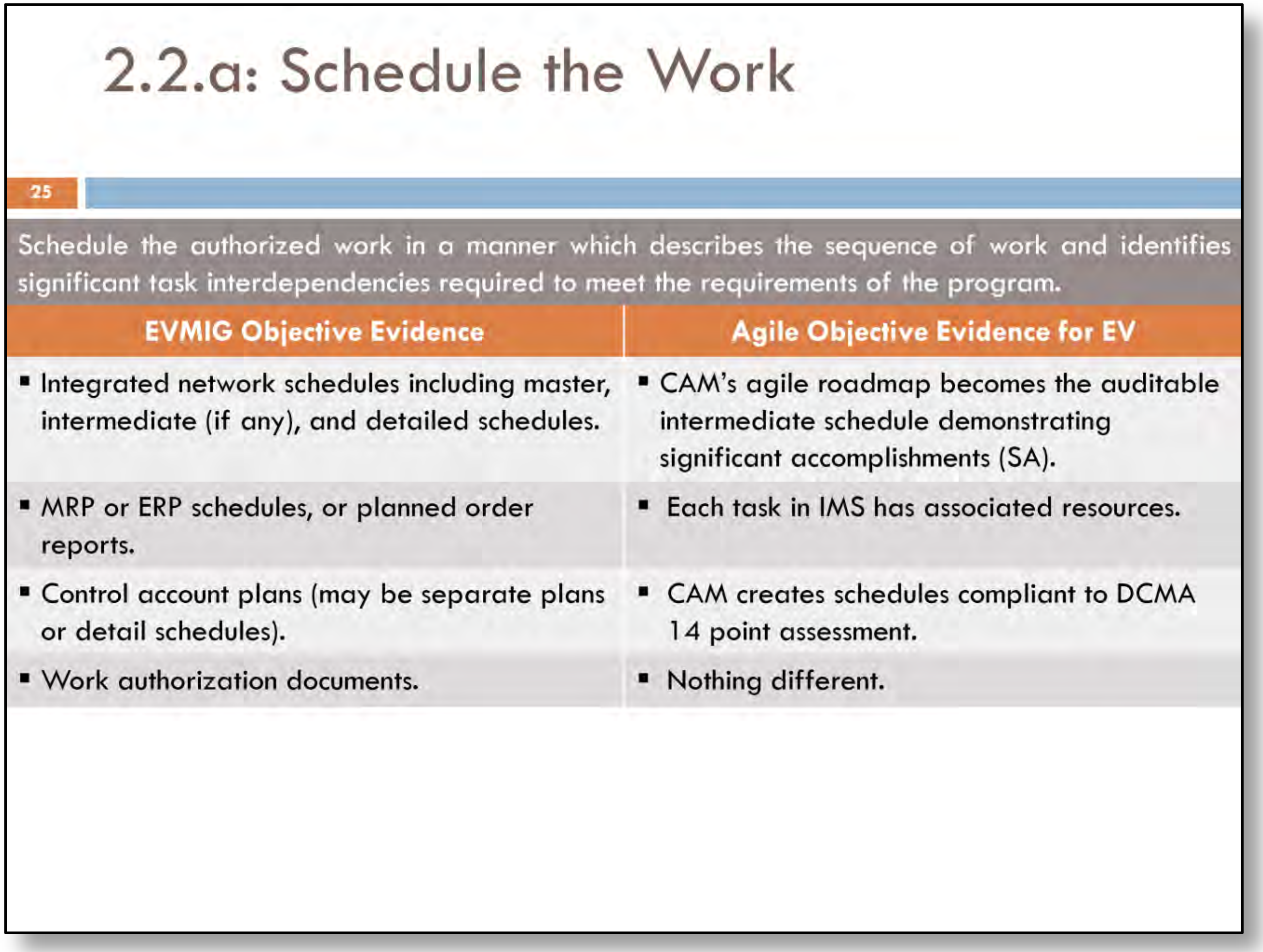
The intersection of the OBS and WBS is the Control Account (CA) with the related Control Account Plan (CAP) and lower level Work Package – Work Authorizations.

This of course is outside the process domain found in Agile, but inside the process domain of Earned Value Management – guide by 748B.

So Agile has contributions to these documents, in ways not normally found on "agile only" projects.

The formality of a Contract Performance Report - DI-MGMT-81466A – is also outside the domain of Agile, but firming connected to DoD programs using Earned Value (> $20M).

Agile provide objective evidence of progress to plan needed to produce the CPR.



2.2.a: Schedule the Work

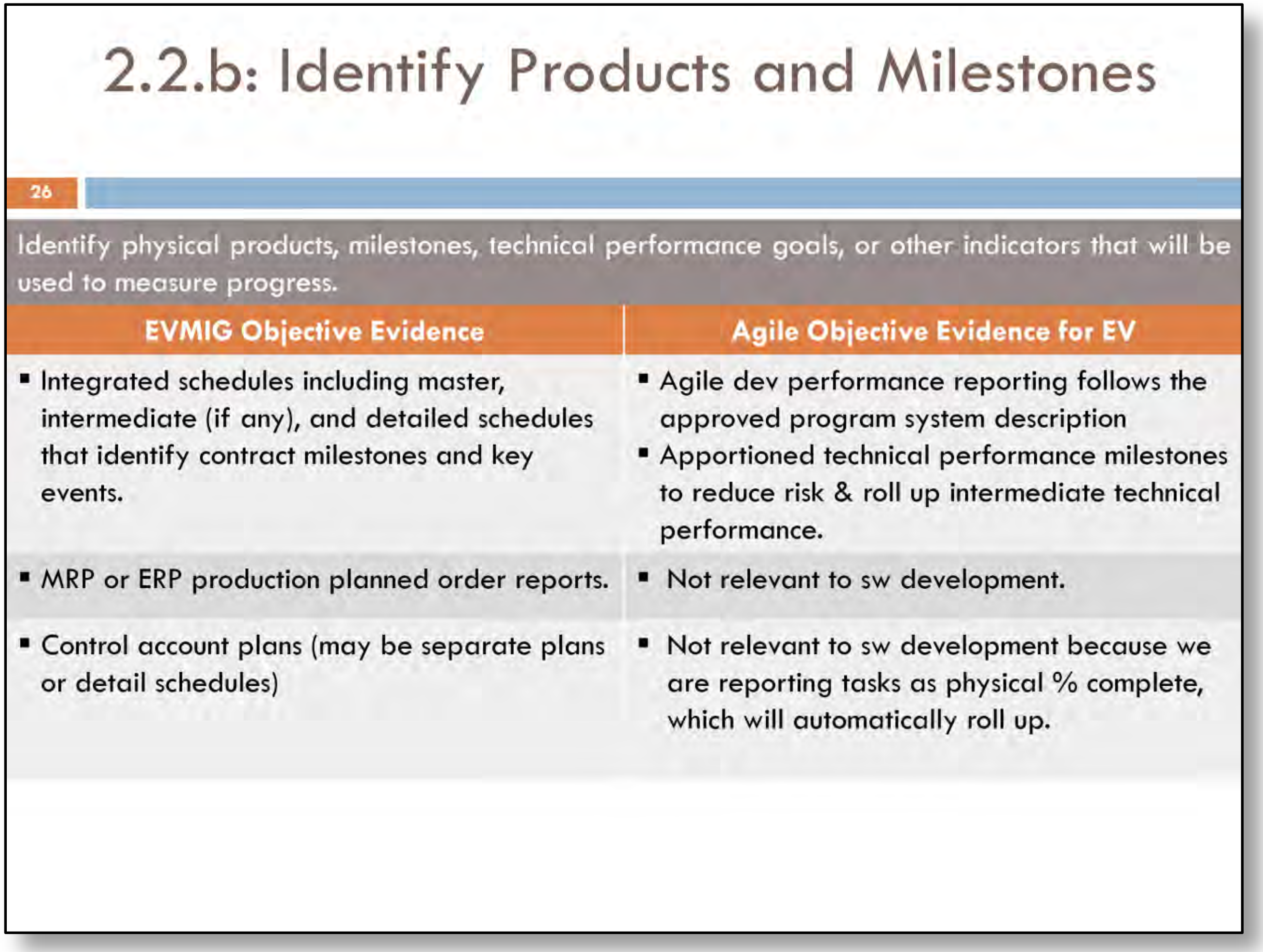
Schedule the authorized work in a manner which describes the sequence of work and identifies significant task interdependencies required to meet the requirements of the program.

EVMIG Objective Evidence	Agile Objective Evidence for EV
▪ Integrated network schedules including master, intermediate (if any), and detailed schedules.	▪ CAM's agile roadmap becomes the auditable intermediate schedule demonstrating significant accomplishments (SA).
▪ MRP or ERP schedules, or planned order reports.	▪ Each task in IMS has associated resources.
▪ Control account plans (may be separate plans or detail schedules).	▪ CAM creates schedules compliant to DCMA 14 point assessment.
▪ Work authorization documents.	▪ Nothing different.

The notion of scheduling in Agile is straight forward. Iterations with fixed durations and fixed staff and a candidate list of features, stories, or other outcomes.

The staff works on fixed boundaries. This is not always possible in integrated programs where software is only part of the deliverable.

So scheduling needs to consider the paradigm of Agile as well as the work processes of the large program.



2.2.b: Identify Products and Milestones

26

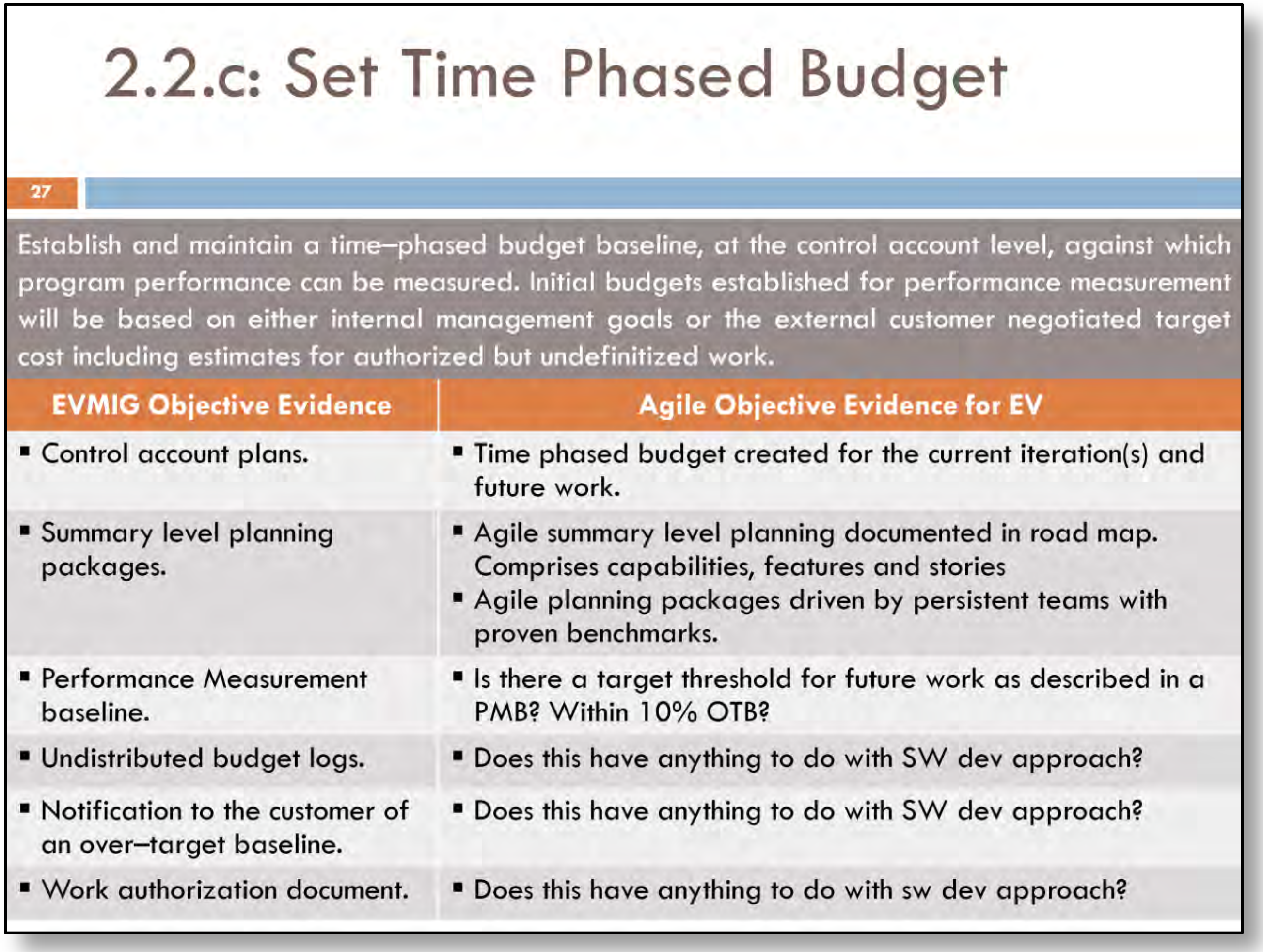
Identify physical products, milestones, technical performance goals, or other indicators that will be used to measure progress.

EVMIG Objective Evidence	Agile Objective Evidence for EV
▪ Integrated schedules including master, intermediate (if any), and detailed schedules that identify contract milestones and key events.	▪ Agile dev performance reporting follows the approved program system description ▪ Apportioned technical performance milestones to reduce risk & roll up intermediate technical performance.
▪ MRP or ERP production planned order reports.	▪ Not relevant to sw development.
▪ Control account plans (may be separate plans or detail schedules)	▪ Not relevant to sw development because we are reporting tasks as physical % complete, which will automatically roll up.

The products and milestones are the assessment points in any program. Agile does this naturally with the defined outcome of “working software” at the end of the iteration.

On large programs this seems to be more difficult. This is the primary reasons for the inclusion of Agile in the IT intensive program development world for DoD.

It forces the discussion of what “done” looks like in terms of tangible working outcomes.



2.2.c: Set Time Phased Budget

27

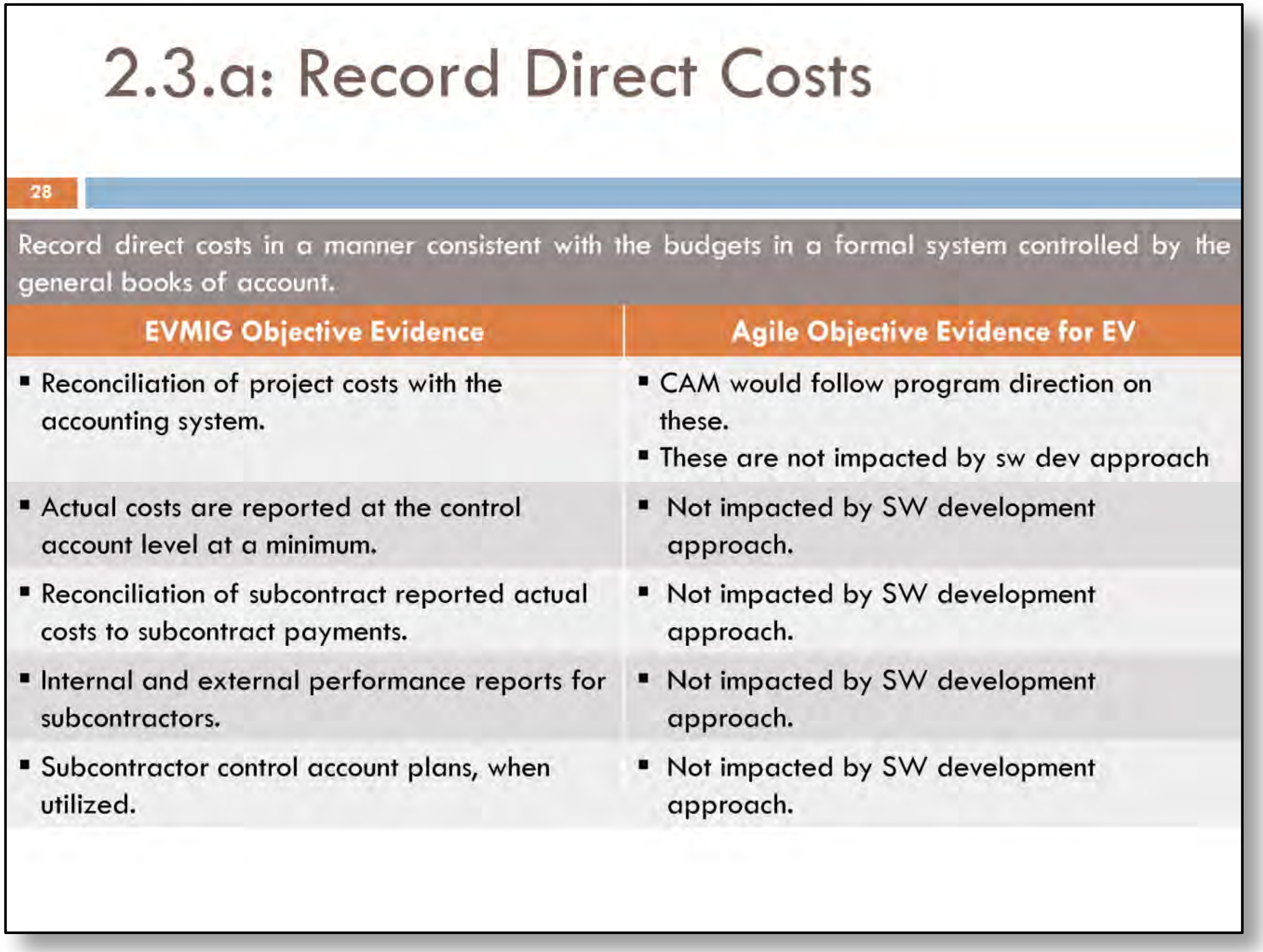
Establish and maintain a time–phased budget baseline, at the control account level, against which program performance can be measured. Initial budgets established for performance measurement will be based on either internal management goals or the external customer negotiated target cost including estimates for authorized but undefinitized work.

EVMIG Objective Evidence	Agile Objective Evidence for EV
▪ Control account plans.	▪ Time phased budget created for the current iteration(s) and future work.
▪ Summary level planning packages.	▪ Agile summary level planning documented in road map. Comprises capabilities, features and stories ▪ Agile planning packages driven by persistent teams with proven benchmarks.
▪ Performance Measurement baseline.	▪ Is there a target threshold for future work as described in a PMB? Within 10% OTB?
▪ Undistributed budget logs.	▪ Does this have anything to do with SW dev approach?
▪ Notification to the customer of an over–target baseline.	▪ Does this have anything to do with SW dev approach?
▪ Work authorization document.	▪ Does this have anything to do with sw dev approach?

Agile has it simple. The time phasing is on fixed boundaries, with nearly fixed expenditures (fixed labor loads), and predefined measures of "done."

On larger EV programs more needs to be done to model the Agile approach. This starts with Technical Performance Measures (TPM) for each deliverable from the Work Package. Traditionally the TPMs were assigned to "large grained" deliverables from the program. The end items, the big chunks of software or metal.

This of course is a mistake and one of the reasons for Agile, to get away from that "big chunk" approach to planning and measuring progress.



2.3.a: Record Direct Costs

28

Record direct costs in a manner consistent with the budgets in a formal system controlled by the general books of account.

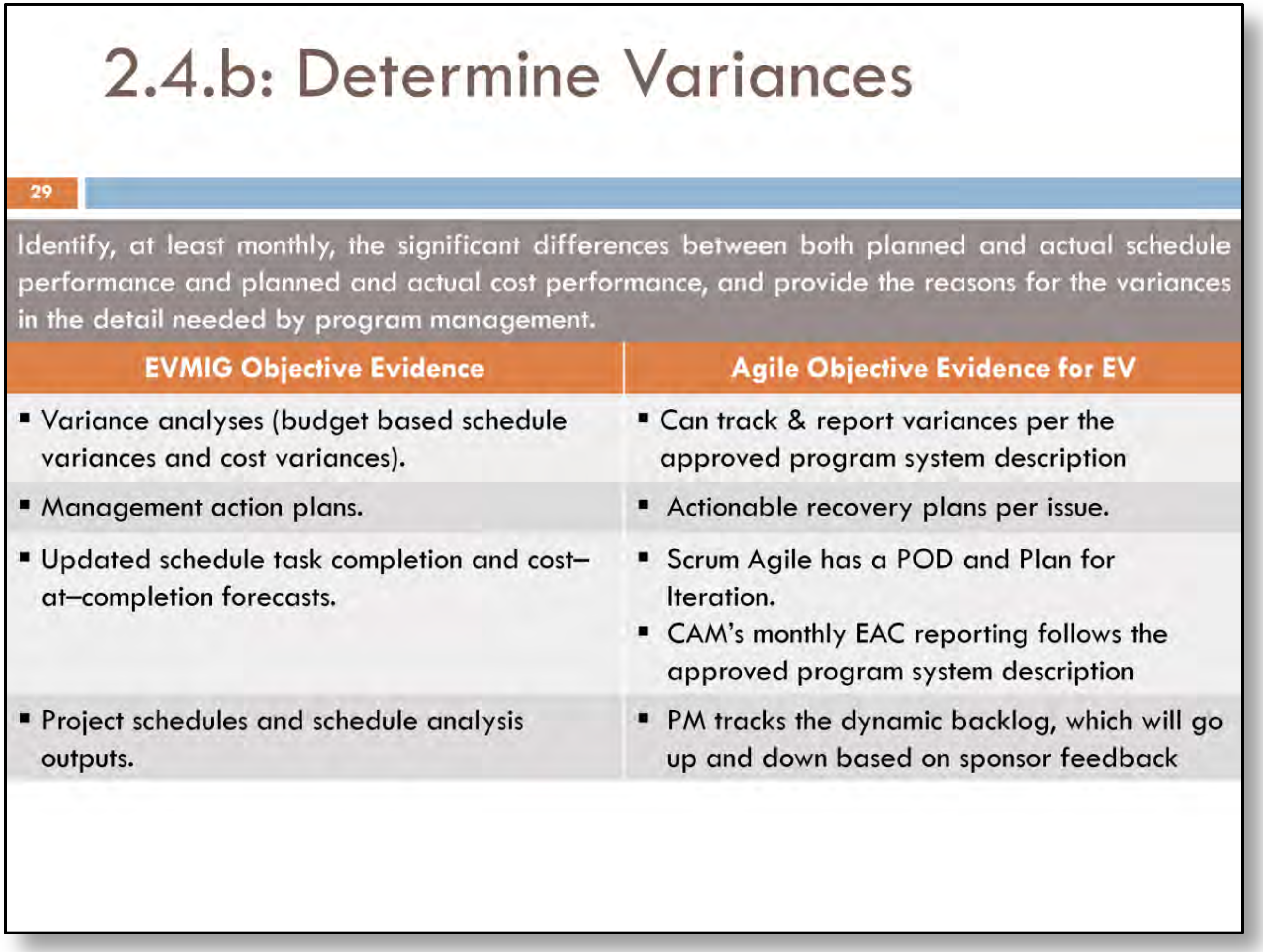
EVMIG Objective Evidence	Agile Objective Evidence for EV
▪ Reconciliation of project costs with the accounting system.	▪ CAM would follow program direction on these. ▪ These are not impacted by sw dev approach
▪ Actual costs are reported at the control account level at a minimum.	▪ Not impacted by SW development approach.
▪ Reconciliation of subcontract reported actual costs to subcontract payments.	▪ Not impacted by SW development approach.
▪ Internal and external performance reports for subcontractors.	▪ Not impacted by SW development approach.
▪ Subcontractor control account plans, when utilized.	▪ Not impacted by SW development approach.

All projects have budgets. No matter is they are Agile or standard EV.

Some form of budget management is needed for all projects.

In more formal projects am accounting system captures and manages these costs. On Agile projects the budget management is straight forward.

The connections between Agile and EV are shown here are simple enough.



2.4.b: Determine Variances

29

Identify, at least monthly, the significant differences between both planned and actual schedule performance and planned and actual cost performance, and provide the reasons for the variances in the detail needed by program management.

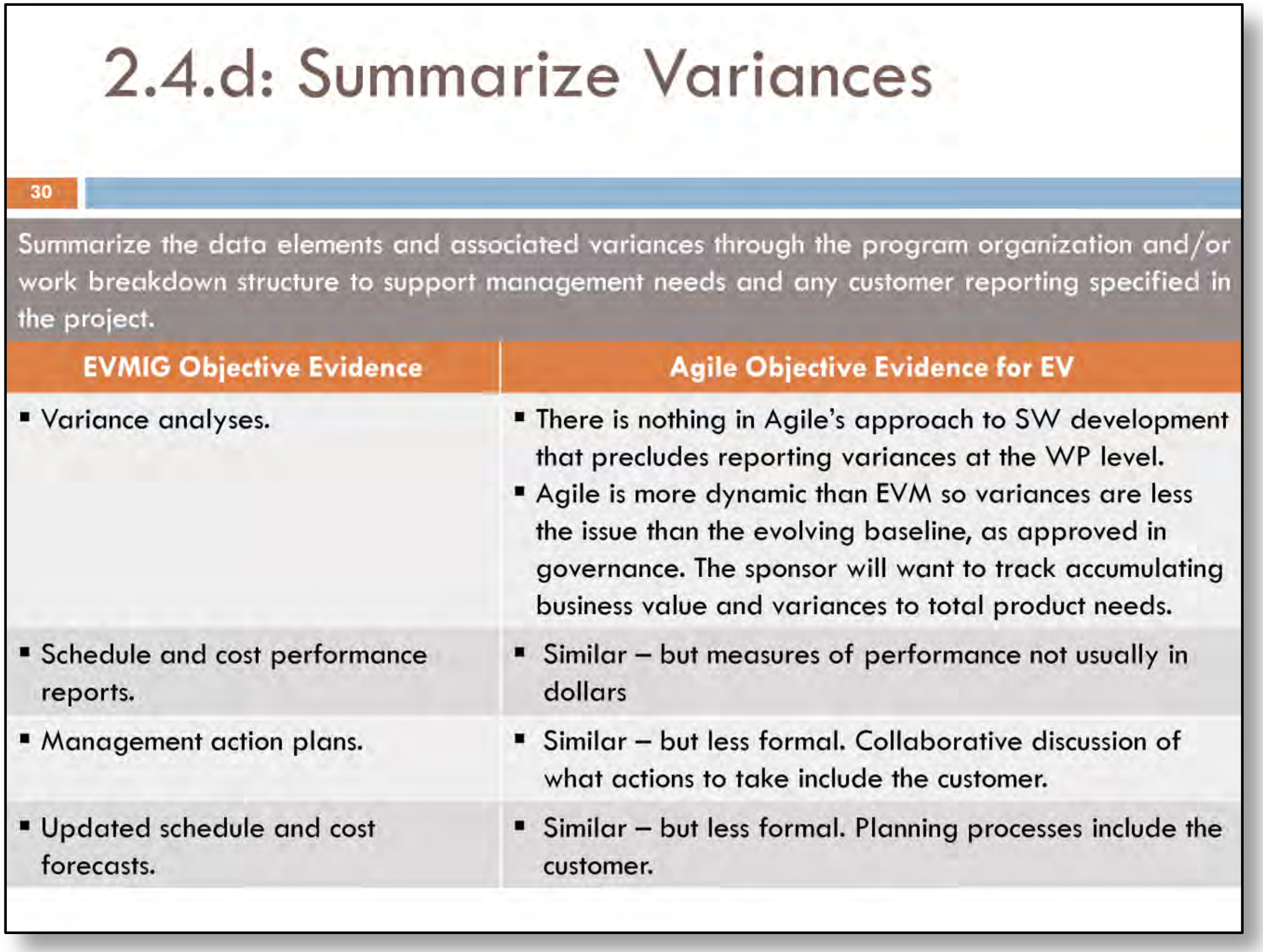
EVMIG Objective Evidence	Agile Objective Evidence for EV
▪ Variance analyses (budget based schedule variances and cost variances).	▪ Can track & report variances per the approved program system description
▪ Management action plans.	▪ Actionable recovery plans per issue.
▪ Updated schedule task completion and cost–at–completion forecasts.	▪ Scrum Agile has a POD and Plan for Iteration. ▪ CAM's monthly EAC reporting follows the approved program system description
▪ Project schedules and schedule analysis outputs.	▪ PM tracks the dynamic backlog, which will go up and down based on sponsor feedback

Again Agile has an advantage here. Fixed iterations with a fixed staff makes capturing actual costs simple.

Not always the case in other paradigms.

No matter what the paradigm, the actual costs – direct cost – needs to be captured in a time phased approach. That is the actual cost capture timeline must be the same as the budgeted baseline cost plan (BCWS).

This is the definition of the Performance Measurement Baseline – a time phased cost.



2.4.d: Summarize Variances

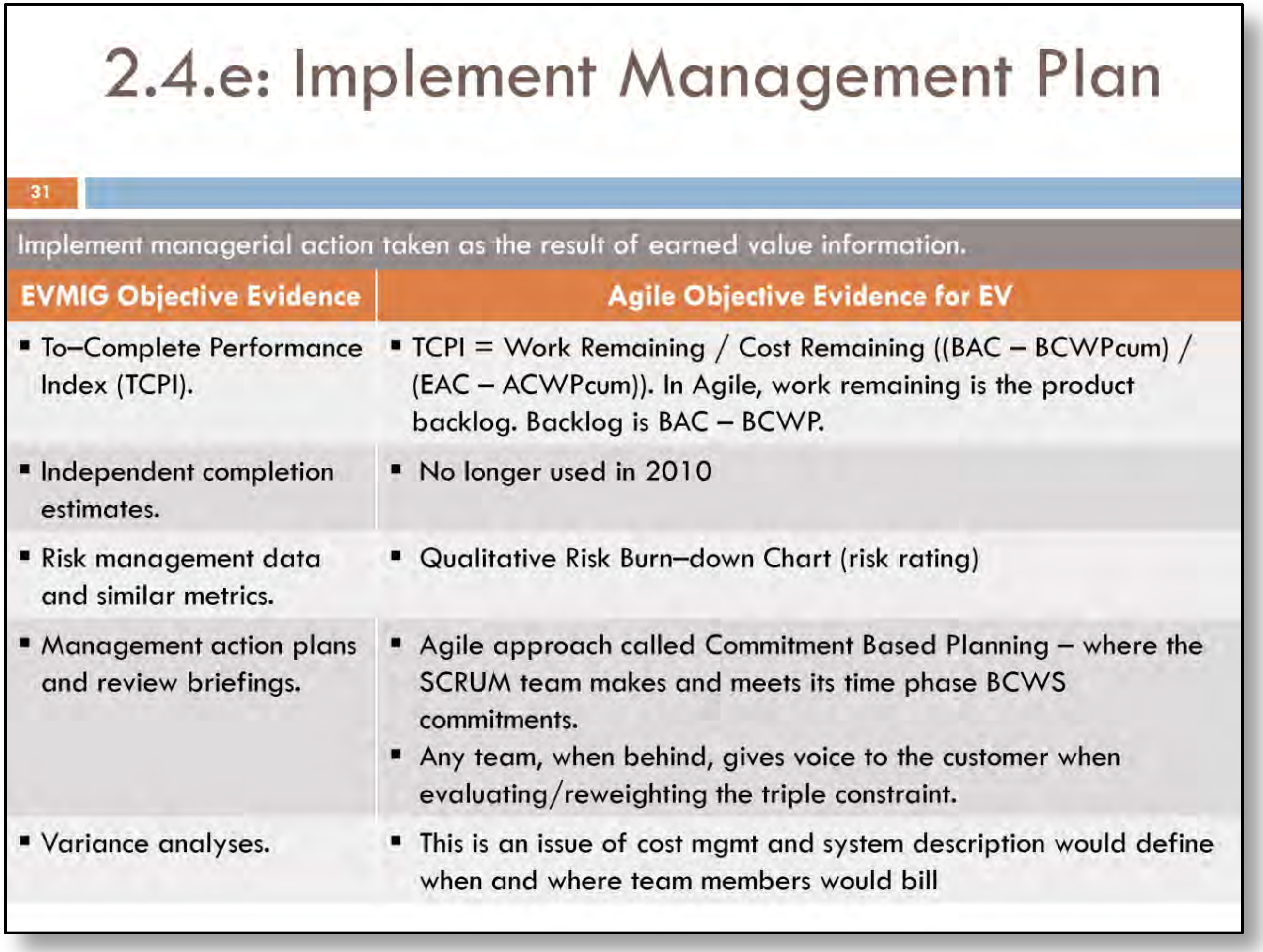
30

Summarize the data elements and associated variances through the program organization and/or work breakdown structure to support management needs and any customer reporting specified in the project.

EVMIG Objective Evidence	Agile Objective Evidence for EV
▪ Variance analyses.	▪ There is nothing in Agile's approach to SW development that precludes reporting variances at the WP level. ▪ Agile is more dynamic than EVM so variances are less the issue than the evolving baseline, as approved in governance. The sponsor will want to track accumulating business value and variances to total product needs.
▪ Schedule and cost performance reports.	▪ Similar – but measures of performance not usually in dollars
▪ Management action plans.	▪ Similar – but less formal. Collaborative discussion of what actions to take include the customer.
▪ Updated schedule and cost forecasts.	▪ Similar – but less formal. Planning processes include the customer.

Summarize means pretty much the same things on Agile and EV. Variances means the same thing too.

As shown here there is nothing in Agile that prevents reporting variances in the same way we do in EV.



2.4.e: Implement Management Plan

31

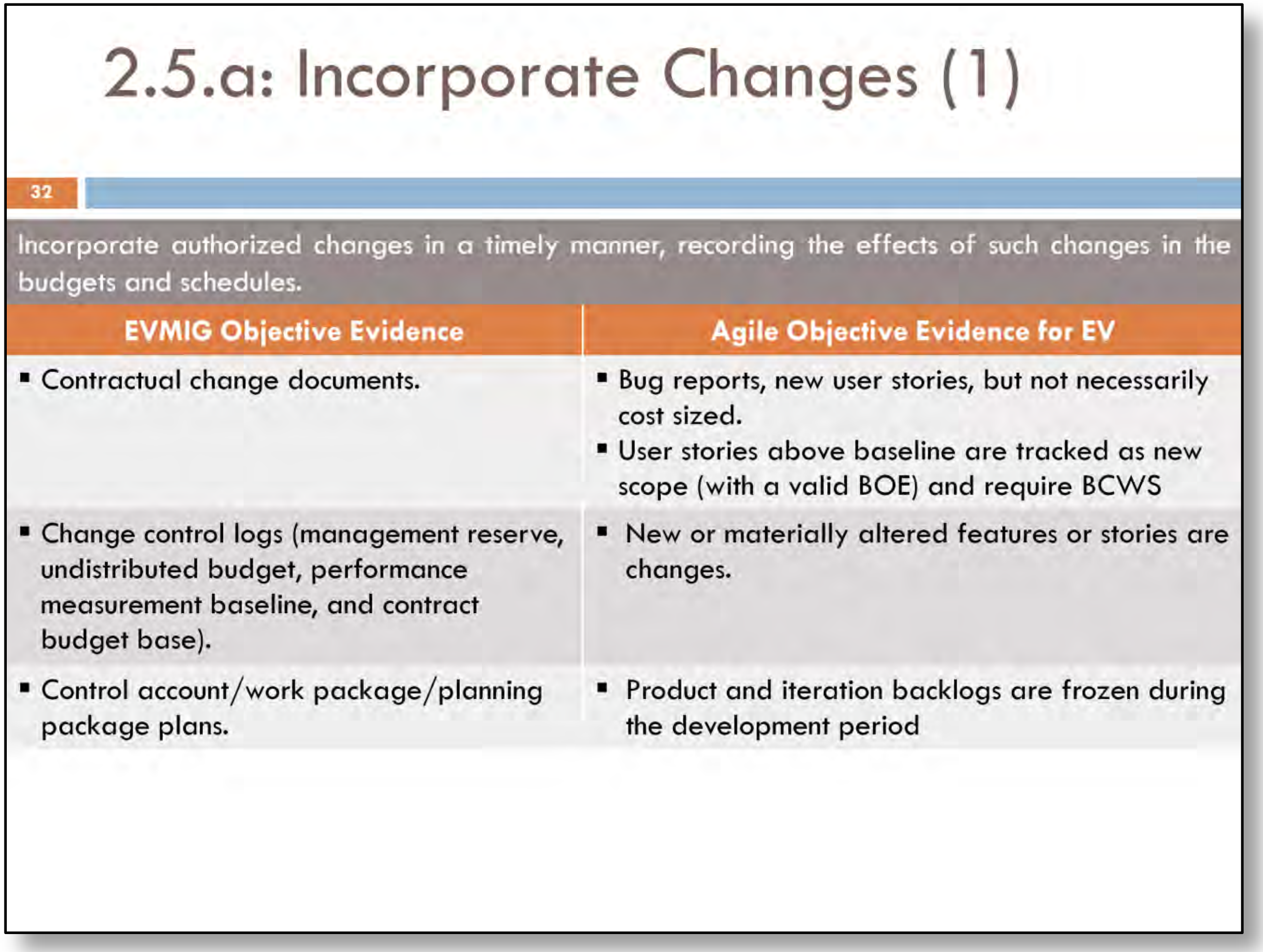
Implement managerial action taken as the result of earned value information.

EVMIG Objective Evidence	Agile Objective Evidence for EV
▪ To–Complete Performance Index (TCPI).	▪ TCPI = Work Remaining / Cost Remaining ((BAC – BCWPcum) / (EAC – ACWPcum)). In Agile, work remaining is the product backlog. Backlog is BAC – BCWP.
▪ Independent completion estimates.	▪ No longer used in 2010
▪ Risk management data and similar metrics.	▪ Qualitative Risk Burn–down Chart (risk rating)
▪ Management action plans and review briefings.	▪ Agile approach called Commitment Based Planning – where the SCRUM team makes and meets its time phase BCWS commitments. ▪ Any team, when behind, gives voice to the customer when evaluating/reweighting the triple constraint.
▪ Variance analyses.	▪ This is an issue of cost mgmt and system description would define when and where team members would bill

Once the variances are determined, management action is needed to make corrections.

In Agile this is not talked about that much. It makes little sense that each Agile iteration completes all the features or stories complete every time.

If they are completed every time, then there might be too much slack in the schedule.



2.5.a: Incorporate Changes (1)

32

Incorporate authorized changes in a timely manner, recording the effects of such changes in the budgets and schedules.

EVMIG Objective Evidence	Agile Objective Evidence for EV
▪ Contractual change documents.	▪ Bug reports, new user stories, but not necessarily cost sized. ▪ User stories above baseline are tracked as new scope (with a valid BOE) and require BCWS
▪ Change control logs (management reserve, undistributed budget, performance measurement baseline, and contract budget base).	▪ New or materially altered features or stories are changes.
▪ Control account/work package/planning package plans.	▪ Product and iteration backlogs are frozen during the development period

When variances appear, corrective action is needed.

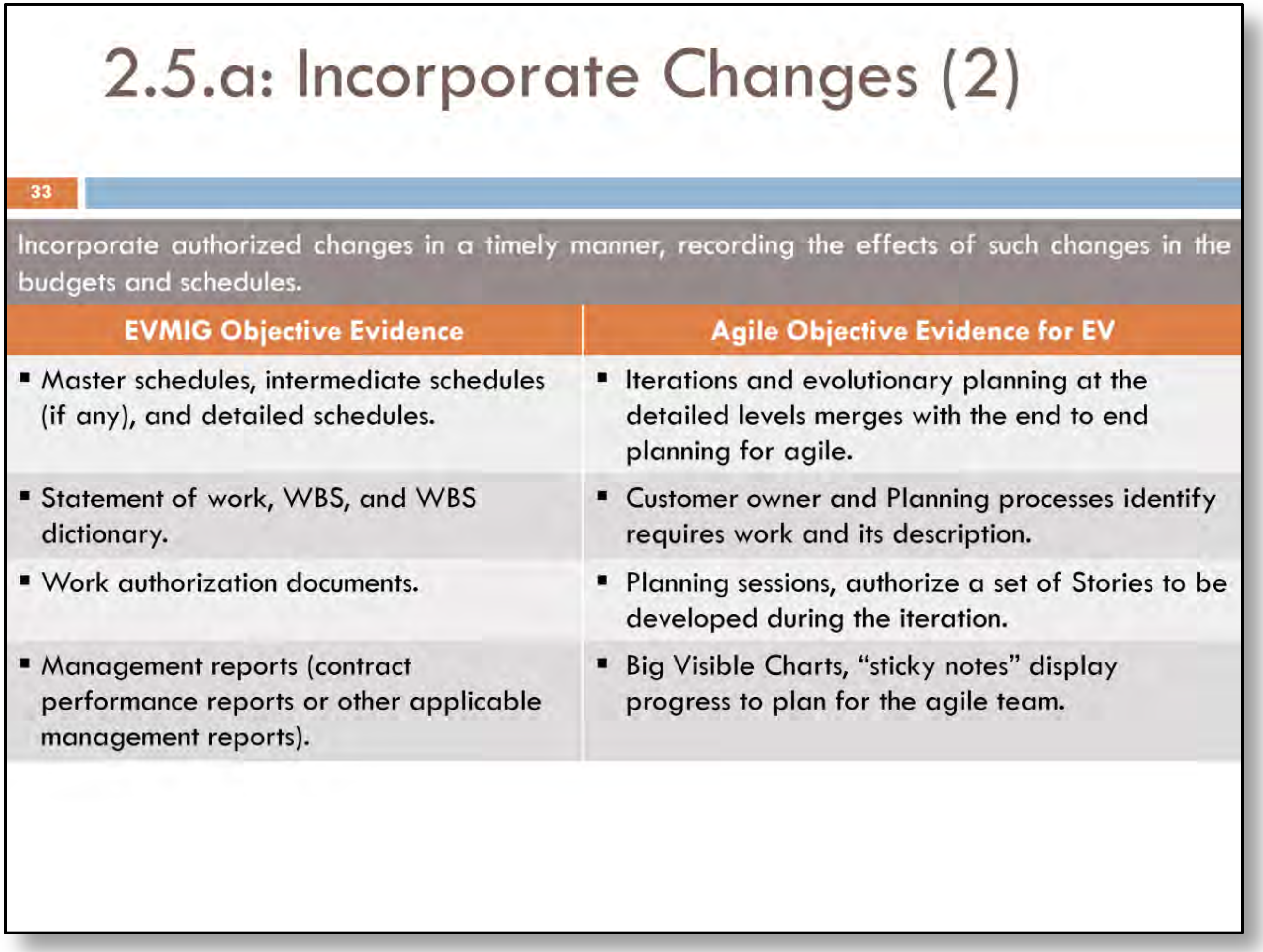
Making changes to the baseline is part of the corrective, after fixing the things that are simply not being right.

But any changes need to be approved, recorded and tracked for compliance.

This is the case in both agile and EV. In agile the formality still needs to be done in some way.

On EV projects the this is mandatory in the ANSI-748B guidelines.

In all cases it’s simply good management.



2.5.a: Incorporate Changes (2)

33

Incorporate authorized changes in a timely manner, recording the effects of such changes in the budgets and schedules.

EVMIG Objective Evidence	Agile Objective Evidence for EV
▪ Master schedules, intermediate schedules (if any), and detailed schedules.	▪ Iterations and evolutionary planning at the detailed levels merges with the end to end planning for agile.
▪ Statement of work, WBS, and WBS dictionary.	▪ Customer owner and Planning processes identify requires work and its description.
▪ Work authorization documents.	▪ Planning sessions, authorize a set of Stories to be developed during the iteration.
▪ Management reports (contract performance reports or other applicable management reports).	▪ Big Visible Charts, “sticky notes” display progress to plan for the agile team.

Making the changes is a management process. Here’s some ways to have it done right in both agile and EV.

Lewis & Fowler
8310 South Valley Highway
Suite 300
Englewood, Colorado 80112
www.lewisandfowler.com
303.524.1610

Deliverables Based Planning®

Integrated Master Plan
Integrated Master Schedule
Earned Value
Risk Management
Proposal Support Service

Glen B. Alleman, VP, Program Planning and Controls
galleman@lewisandfowler.com
303.437 5226

NDIA
National Defense Industrial Association

Information Systems Summit II (ISS II)

sponsored by

C4ISR Division

Baltimore, Md
April 4-6, 2011

ISS II Introduction

Hyatt Regency Baltimore, Md
April 4-6, 2011

Dr Steven Kimmel
Alion Science and Technology
C4ISR Division & Program Chair

ISS II Program Summary

- Commercial Experience with Agile Software Development & Test
- Information Sharing, Assurance and User expectations
- IT policy, management and CIO imperatives to deliver Information System solutions

ISS History
ISS I convened Jan 2009

Improving Defense
Information System
(IS)
Acquisition**:**

Testing IS Capability in a Network Enterprise

DOD IT is a National “Asset”

- National security industrial base key components
  - C4 (decision support, command execution, information-sharing, connectivity, and situational understanding)
  - ISR (means and tools which generate the data)
  - IT (provides infrastructure and support)
- Directly tied to war fighting
  - Defend the networks and systems
  - Attack our enemies networks and systems
- Network-enabled C4ISR/IT capabilities
  - On the ground situational awareness
  - Enhances our warfighting capabilities

Stakeholder Recommendations

- Defense Science Board
  - Developmental Test and Evaluation, May 2008
  - DOD Policies and Procedures for the Acquisition of Information Technology, March 2009

- 2010 National Defense Authorization Act (Sec 804)
  - Implementation of a New Acquisition Process for Information Technology Systems

- NDIA-OUSD (AT&L) System Engineering Division/Development T&E Committee and Software Industry Experts
  - Software T&E Summit/Workshop September 2009
  - Joint Authored White Paper, Dec 2009

- National Academies of Science Study
  - Achieving Effective Acquisition of Information Technology in the Department of Defense, December 2009

New Emerging DOD IT Landscape

- **804 Compliance**: FY2010 National Defense Authorization Act, Section 804
  - New IT system acquisition process required
    - Multiple rapid capability increment
    - Early prototyping for evolutionary approach

- **Early & continual user involvement**
- **Multiple, rapidly executed capability increments or releases**
- **Early, successive prototyping to support evolutionary approach**
- **Modular, open-systems approach**

DOD Agile IT Precepts

1) achieve significant time and cost resource efficiencies

2) support software application “sprints”

3) provide tailored test environments established on demand

4) create a virtual library of systems and services to avoid having to stand up physical systems for every test

5) establish a DOD wide accepted restructured IT T&E process

NDIA National Defense Industrial Association

DOD Agile IT Challenges

- *Incremental Requirements (IT Box)*
- *Incremental Budgets*
- *Iterative S&T Investments*
- *Iterative Systems Engineering*

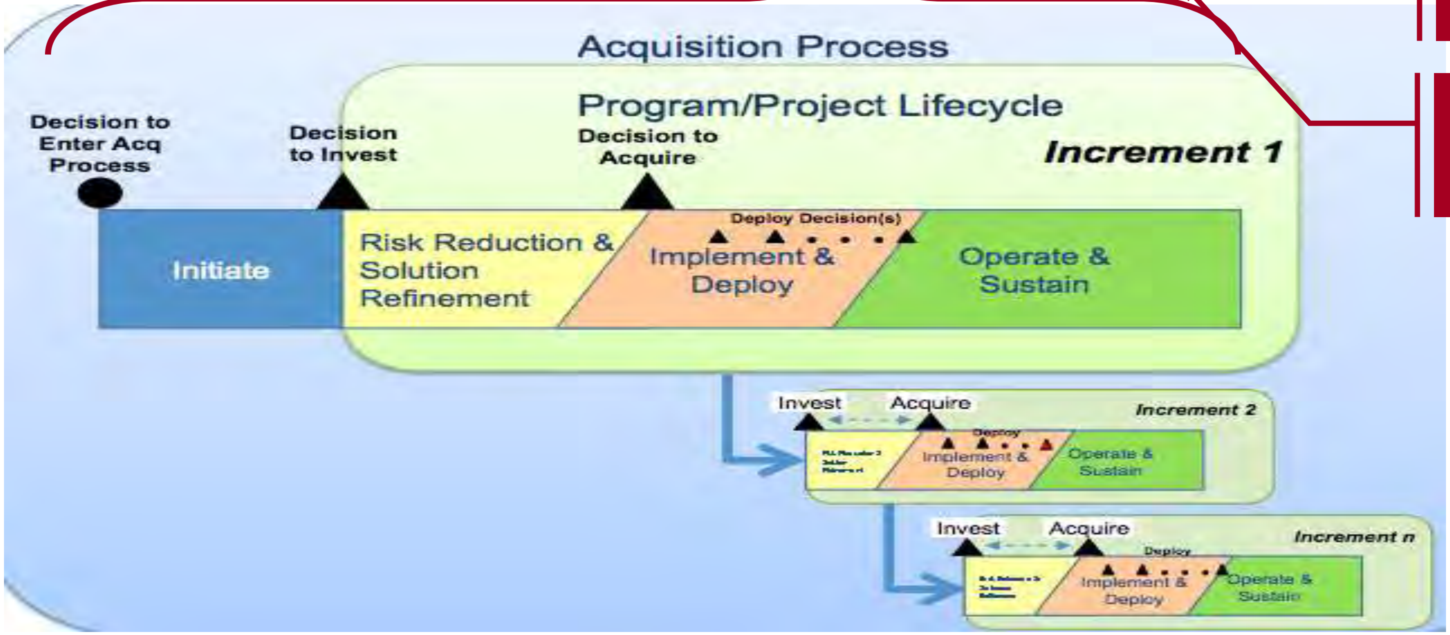
FRCB ??

Testing ??

C&A ??

Contract Language?

Warfighter Feedback?



Closing Comments

- IT systems are different than weapons systems…current DOD 5000 inappropriate for both

- DOD agile developed IT systems/applications are on the horizon……IT "sprints" projects will cause closer collaboration between users, developers and testers

- Serial or "soda straw" T&E to migrate towards –
  - parallel acceptance, certification, accreditation, interoperability and integration
  - test once, accept by all.

Information Systems Summit II (ISS II)

sponsored by

C4ISR Division

Baltimore, Md
April 4-6, 2011

NDIA
National Defense Industrial Association

Information Systems Summit II (ISS II)

sponsored by

C4ISR Division

Baltimore, Md
April 4-6, 2011

ISS II Introduction

Hyatt Regency Baltimore, Md
April 4-6, 2011

Dr Steven Kimmel
Alion Science and Technology
C4ISR Division & Program Chair

ISS II Program Summary

- Commercial Experience with Agile Software Development & Test
- Information Sharing, Assurance and User expectations
- IT policy, management and CIO imperatives to deliver Information System solutions

ISS History
ISS I convened Jan 2009

Improving Defense
Information System
(IS)
Acquisition**:**

Testing IS Capability in a Network Enterprise

DOD IT is a National “Asset”

- National security industrial base key components
  - C4 (decision support, command execution, information-sharing, connectivity, and situational understanding)
  - ISR (means and tools which generate the data)
  - IT (provides infrastructure and support)
- Directly tied to war fighting
  - Defend the networks and systems
  - Attack our enemies networks and systems
- Network-enabled C4ISR/IT capabilities
  - On the ground situational awareness
  - Enhances our warfighting capabilities

Stakeholder Recommendations

- Defense Science Board
  - Developmental Test and Evaluation, May 2008
  - DOD Policies and Procedures for the Acquisition of Information Technology, March 2009
- 2010 National Defense Authorization Act (Sec 804)
  - Implementation of a New Acquisition Process for Information Technology Systems
- NDIA-OUSD (AT&L) System Engineering Division/Development T&E Committee and Software Industry Experts
  - Software T&E Summit/Workshop September 2009
  - Joint Authored White Paper, Dec 2009
- National Academies of Science Study
  - Achieving Effective Acquisition of Information Technology in the Department of Defense, December 2009

New Emerging DOD IT Landscape

- **804 Compliance**: FY2010 National Defense Authorization Act, Section 804
  - New IT system acquisition process required
    - Multiple rapid capability increment
    - Early prototyping for evolutionary approach

- **Early & continual user involvement**
- **Multiple, rapidly executed capability increments or releases**
- **Early, successive prototyping to support evolutionary approach**
- **Modular, open-systems approach**

DOD Agile IT Precepts

1) achieve significant time and cost resource efficiencies

2) support software application “sprints”

3) provide tailored test environments established on demand

4) create a virtual library of systems and services to avoid having to stand up physical systems for every test

5) establish a DOD wide accepted restructured IT T&E process

DOD Agile IT Challenges

- *Incremental Requirements (IT Box)*
- *Incremental Budgets*
- *Iterative S&T Investments*
- *Iterative Systems Engineering*

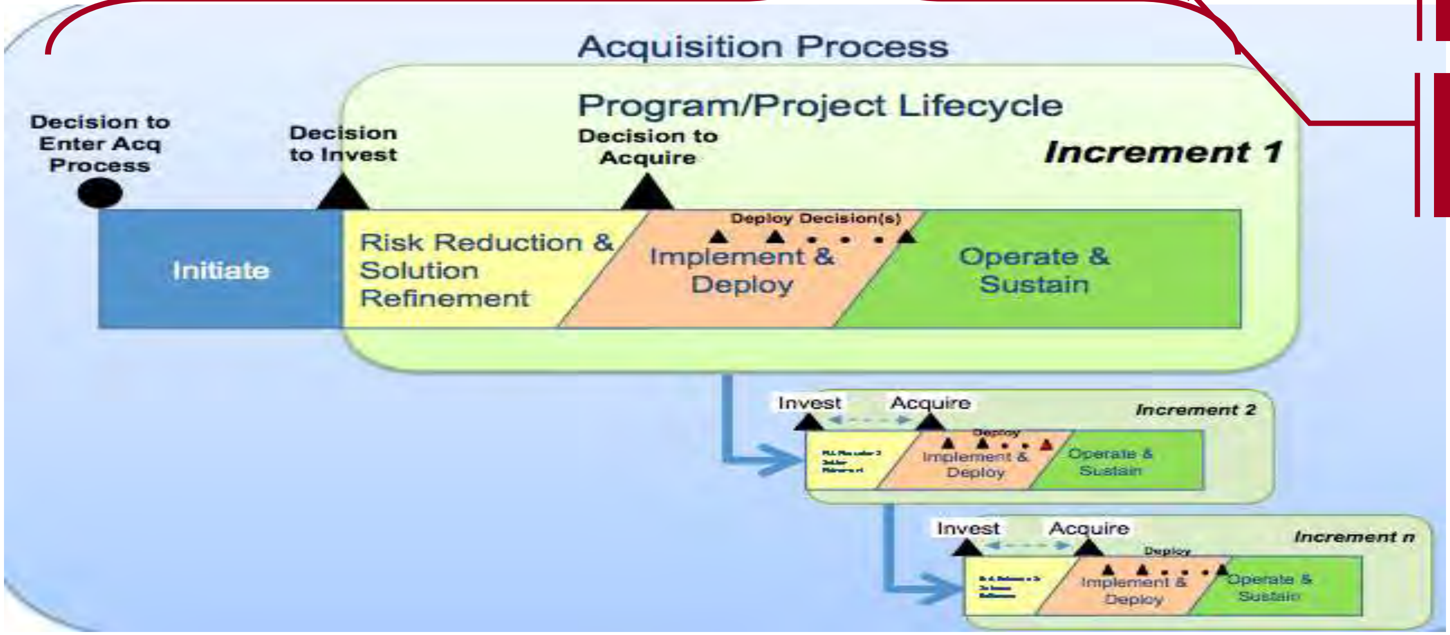
FRCB ??

Testing ??

C&A ??

Contract Language?

Warfighter Feedback?



Closing Comments

- IT systems are different than weapons systems…current DOD 5000 inappropriate for both
- DOD agile developed IT systems/applications are on the horizon……IT "sprints" projects will cause closer collaboration between users, developers and testers
- Serial or "soda straw" T&E to migrate towards –
  - parallel acceptance, certification, accreditation, interoperability and integration
  - test once, accept by all.

Information Systems Summit II (ISS II)

sponsored by

C4ISR Division

Baltimore, Md
April 4-6, 2011

Lean

The Role of the Product Owner

Agile

Michael Cox

Vice President and Senior Consultant

NetObjectives, Inc.

michael.cox@netobjectives.com

“What is all this Agile stuff about, anyway?”

The Role Of the Product Owner

The traditional role

Why change?

What does the agile role look like?

Examples

The Traditional Role

- Multiple product owners and business stakeholders provide input and define requirements
- Sponsors often high in the organization – funding the project but not into the details
- Competing decision makers – i.e. IT and Business
- Mostly involved in the front end requirements and backend tests
- Receive status from program managers

The Traditional Role

- Detailed plans are put together up front
- Progress toward achieving desired product is based on compliance with a plan
- Management of tasks via status meetings
- Utilization of resources – especially people
- Command and control to tell the team what to work on and define due dates (often in conflict)

The Traditional Role – the Issue

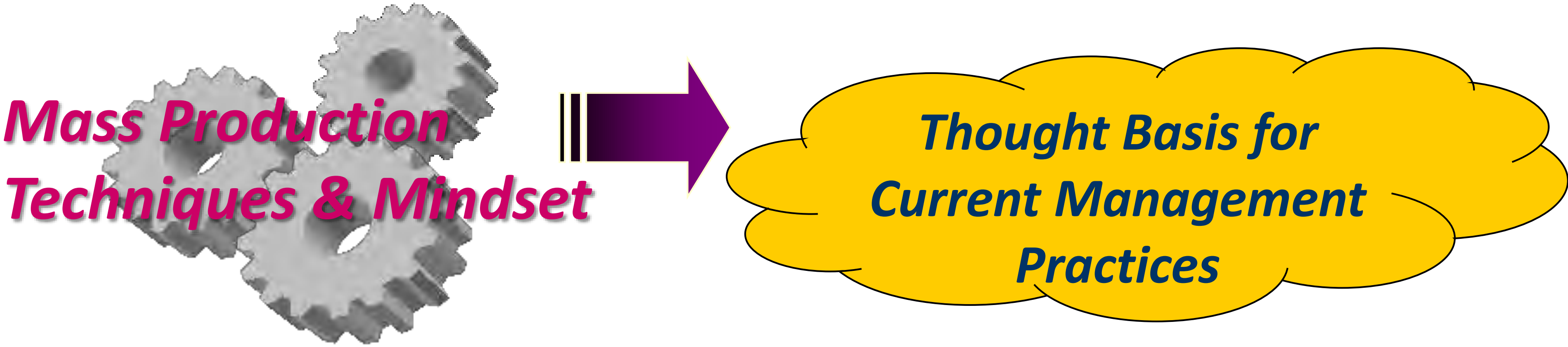
- Conflict between trying to define requirements *a priori* and time to market (or cycle time)
- Customer and market needs are brought in too late
- Product does not meet customer's needs (cost, schedule, functionality)
- Amplified within the DoD where the acquisition customer and the end customer are not the same

How Did We Get Into This Spot?

- Tremendous rise in the standard of living the past 100 years in all developed countries
- Rise was largely driven by productivity improvements
  - Agricultural up 3 to 5% a year since 1900
    - 50% of workforce in 1900, < 2% today, more production
  - Production up by 3% a year since Depression
    - 35% of workforce in 1940, < 15% today, 100x output rise

Basis has been the Invention and Widespread Adoption of Mass Production Techniques

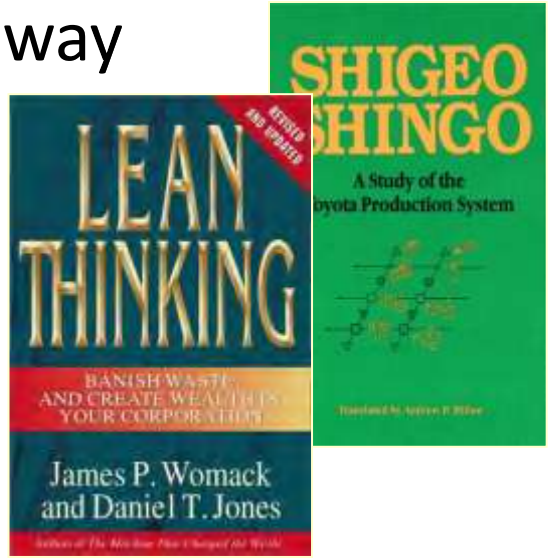
How Did We Get Into This Spot?



Mass Production Techniques & Mindset

Thought Basis for Current Management Practices

- Managing via hierarchy, command and control
- Scientific management – the one best way
- Economies of scale
- Batch production


LEAN THINKING
REVISED AND UPDATED
BANISH WASTE AND CREATE WEALTH IN YOUR CORPORATION
James P. Womack and Daniel T. Jones
SHIGEO SHINGO
A Study of the Toyota Production System

Lean Principles have generated Lean Practices

How Did We Get Into This Spot?

- Mass production management techniques in systems and software development have largely failed
  - Documentation = Understanding
  - The right tasks, correct pressure - force it to happen
  - “If they would freeze requirements, we would be fine”
  - “Heroes” called in when program is in real trouble
- A dissatisfied customer community has imposed more controls and rigidity
- Contractors countered with rigid contracts and change orders to batter the customer with cost and schedule
- Product owners were not involved until too late

we are always working with uncertainty

Requirements …

Decay and

Lose Value

over time

Requirements

are not fully understood even after a formal sign-off

Requirements

change often

during long development cycles

Requirements

piled on

poorly prioritized

long delivery cycles

What does Agile demand from Process Standpoint?

What does Agile demand from the Product Owner?

Agility

Predictability

of Business Value Realization

Agile

Agile is about

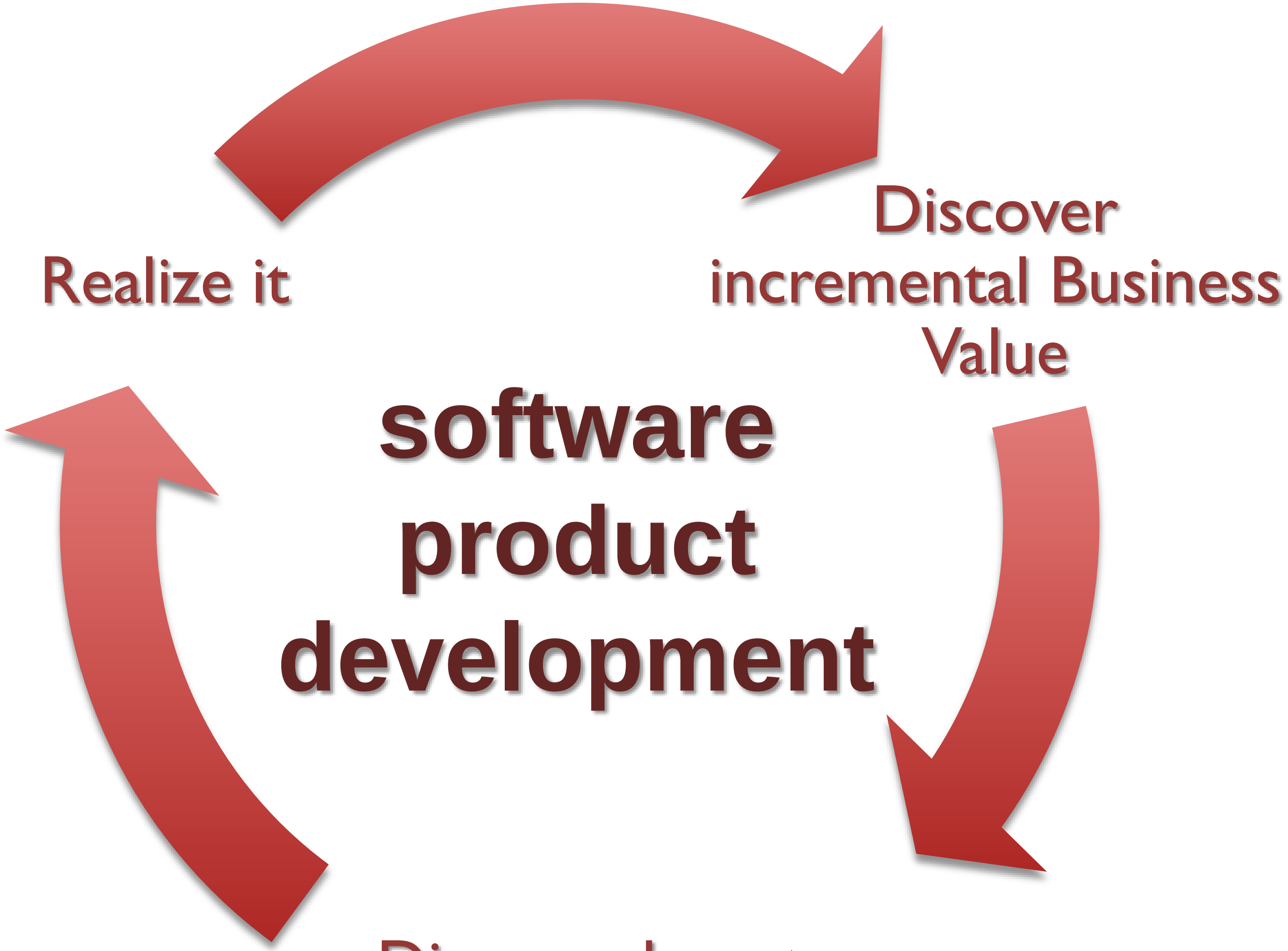
Business Iterations
not Development
Cycles

Agile

Agile is a method that features rapid delivery of functional product iterations

Relies on immediate customer feedback

Allows for evolving understanding of system



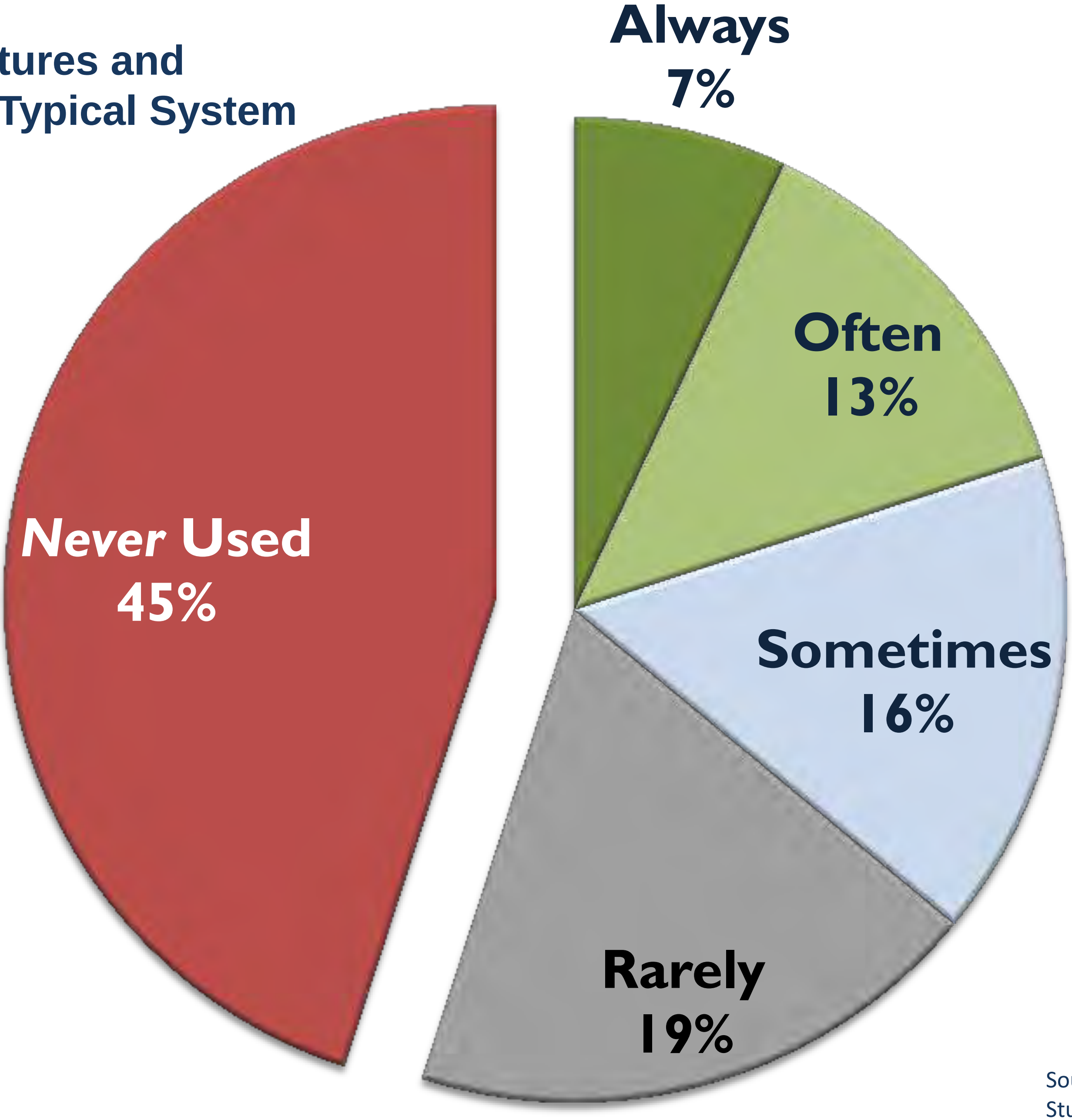
Realize it

Discover
incremental Business
Value

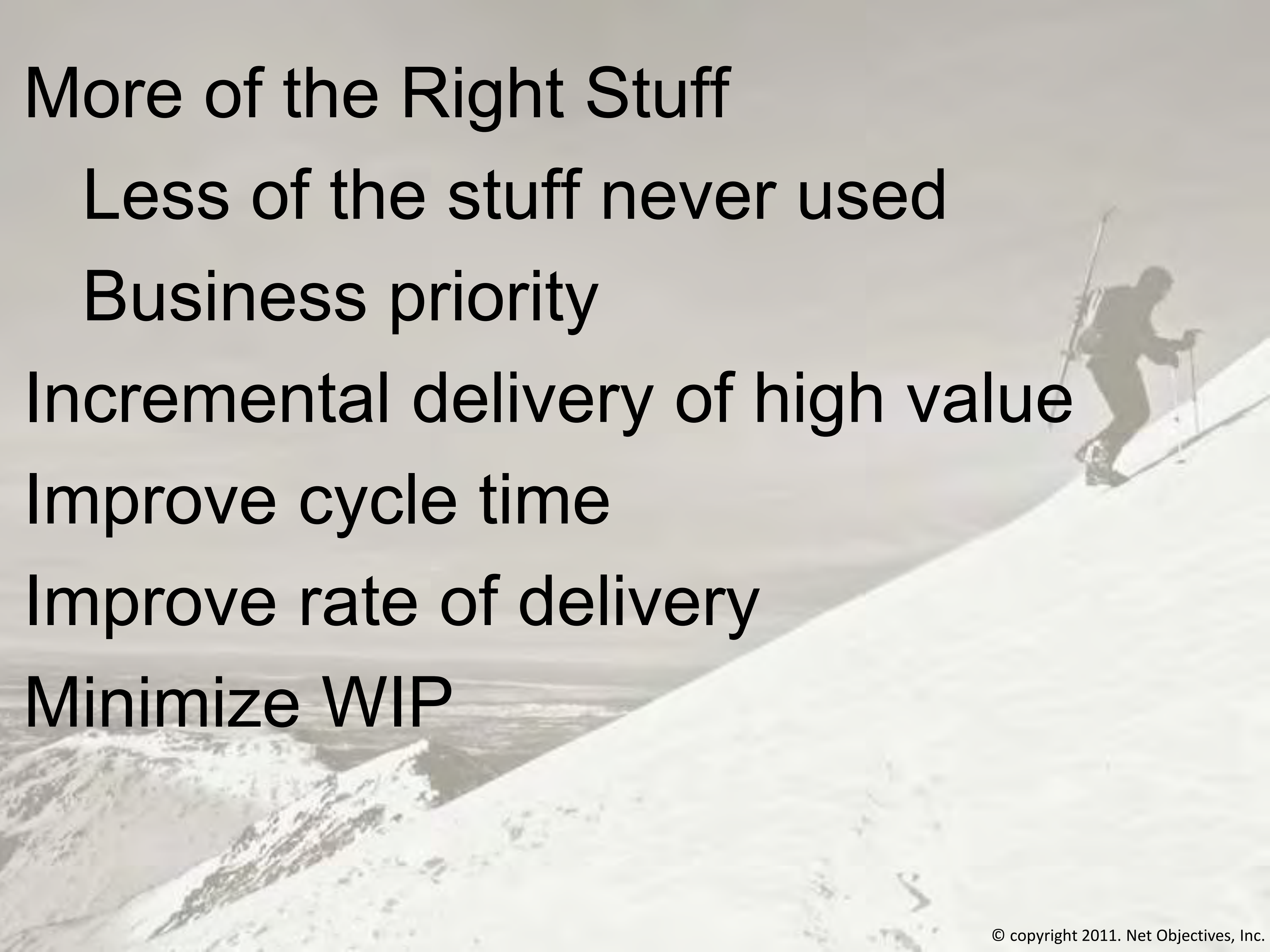
**software
product
development**

Discover how to
build & implement it

Usage of Features and Functions in Typical System



Source: Standish Group Study of 2000 projects at 1000 companies



- More of the Right Stuff
  - Less of the stuff never used
  - Business priority
- Incremental delivery of high value
- Improve cycle time
- Improve rate of delivery
- Minimize WIP

“The greatest improvement in knowledge work will come from simply not doing what does not need to be done”

Peter F. Drucker

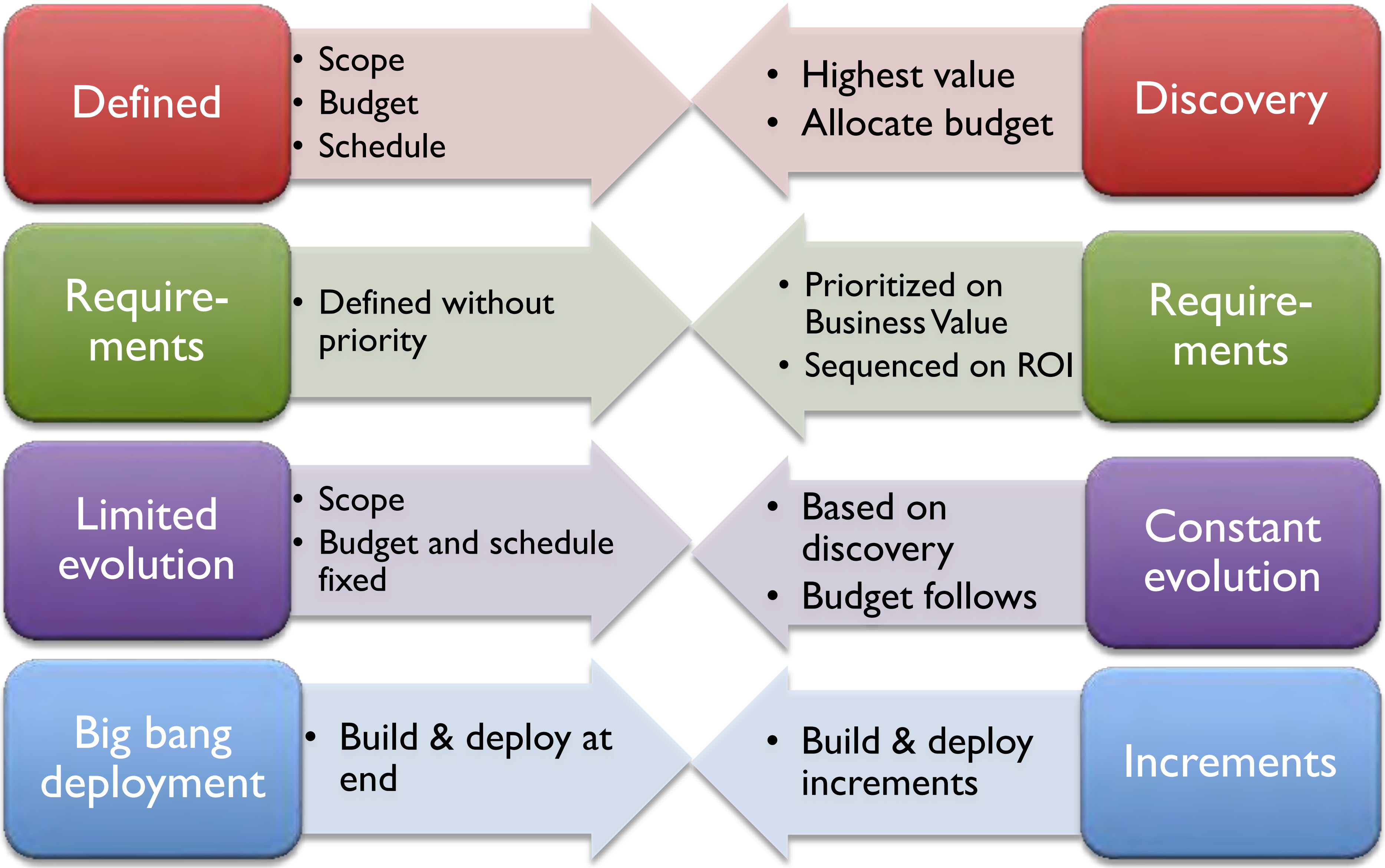
Harvard Business Review
“The New Productivity Challenge”
November/December 1991

“You cannot build the right thing if you have not discovered it first!”

This is the role of the product owner in agile development!

Project-based

Business Value-based



Defined

- Scope
- Budget
- Schedule

Discovery

- Highest value
- Allocate budget

Require-ments

- Defined without priority

Require-ments

- Prioritized on Business Value
- Sequenced on ROI

Limited evolution

- Scope
- Budget and schedule fixed

Constant evolution

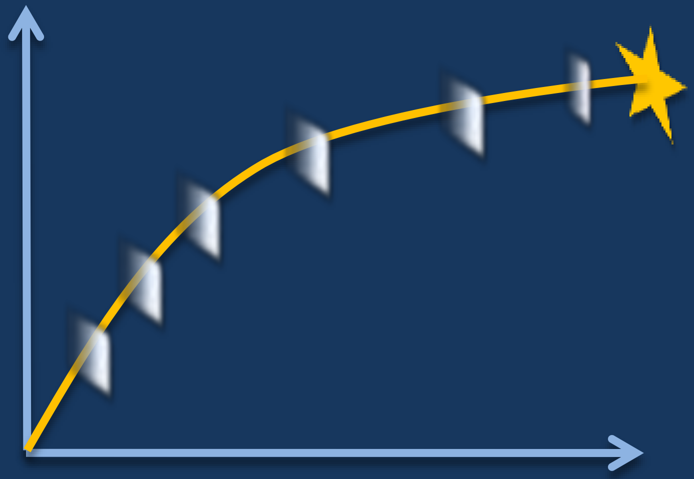
- Based on discovery
- Budget follows

Big bang deployment

- Build & deploy at end

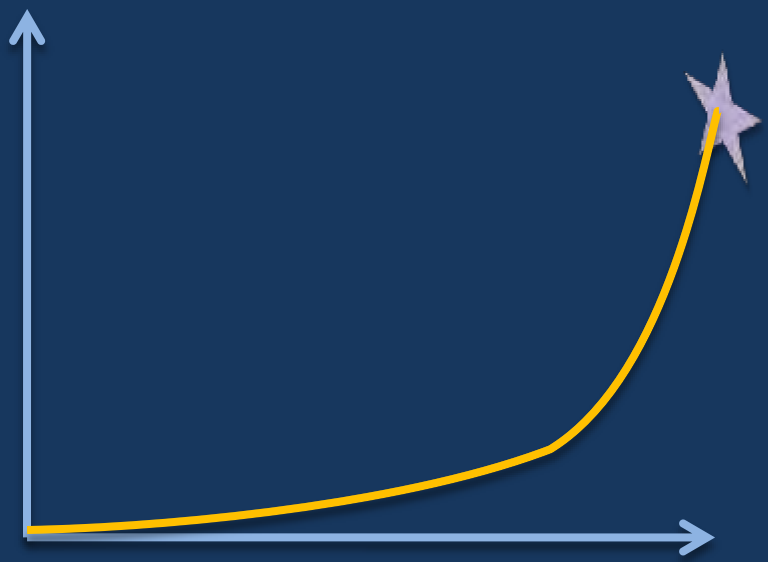
Increments

- Build & deploy increments

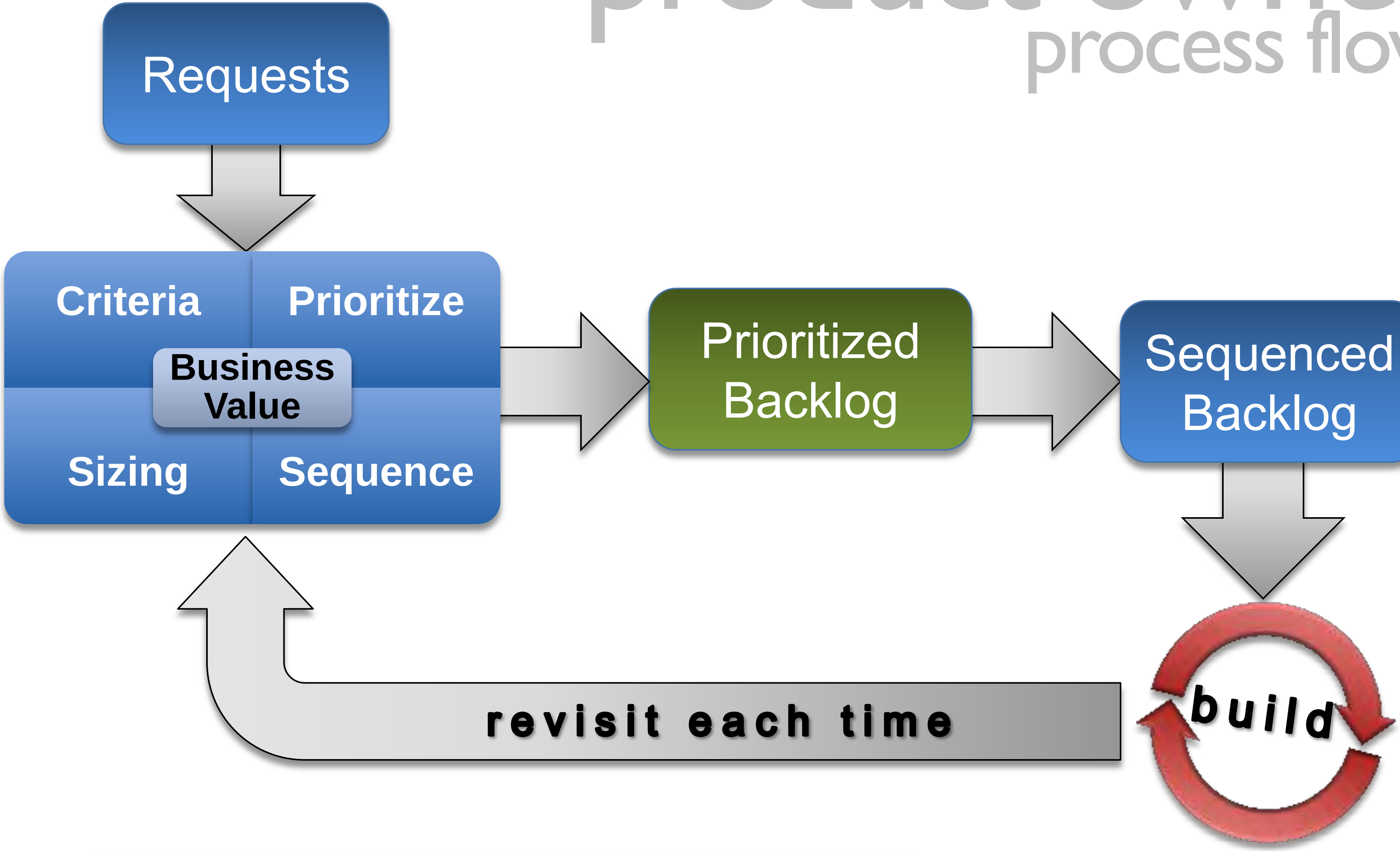


Incrementally Realizing Business Value

Evolving the System



product owner
process flow



Requests

Criteria

Prioritize

Business Value

Sizing

Sequence

Prioritized Backlog

Sequenced Backlog

build

revisit each time

Product Owner Must Drive the Process

Role of Business Product Owner

- Creates and maintains the Product Backlog
- Prioritizes and sequences the Backlog according to business value or ROI
- Assists with the breakdown of Features into user stories that are granular enough to build quickly
- Conveys the Vision and Goals at the beginning of every Release and Sprint

Role of Business Product Owner

- Represents the customer, interfaces and engages the customer
- Participates in the daily meetings of the team
- Responsible for buyoff of the incremental product progress
- Has responsibility to define when work is done and complete authority to accept or reject it

Role of Business Product Owner

- Ability to manage dependencies and risks
- Ability to prioritize and sequence business needs
- Deep understanding of what the customer needs
- Good intuition of the development team's capabilities
- Unafraid to set direction for the product without telling the team how to develop it

Product Owner – the Agile Reality

- Can no longer be hands off
- Can not simply write requirements and then take delivery
- Must continuously drive for incremental realization of valuable product
- Must remove impediments

Responsibilities of a Product Owner / Customer

- Determine what Stakeholders Want
- Decide what They Actually Get
- Drive the Team at a Sustainable Pace
- Write Stories Representing This
- Explain The Stories to the Team
- Approve the Functional Tests
- Validate That We Got What We Wanted
- Release the Product

The Product Owner

Must pay attention to all the ‘stories’ within a feature

- User Story (Business Functionality – value)
- Analysis (discover what to build / How to build it)
- Development Story (system capability)
- Enabling (ex. Training, tools, process)
- Change Mgmt (how the value will be launched & used)

And also at Release and Product Levels (and Portfolio)

…AND…

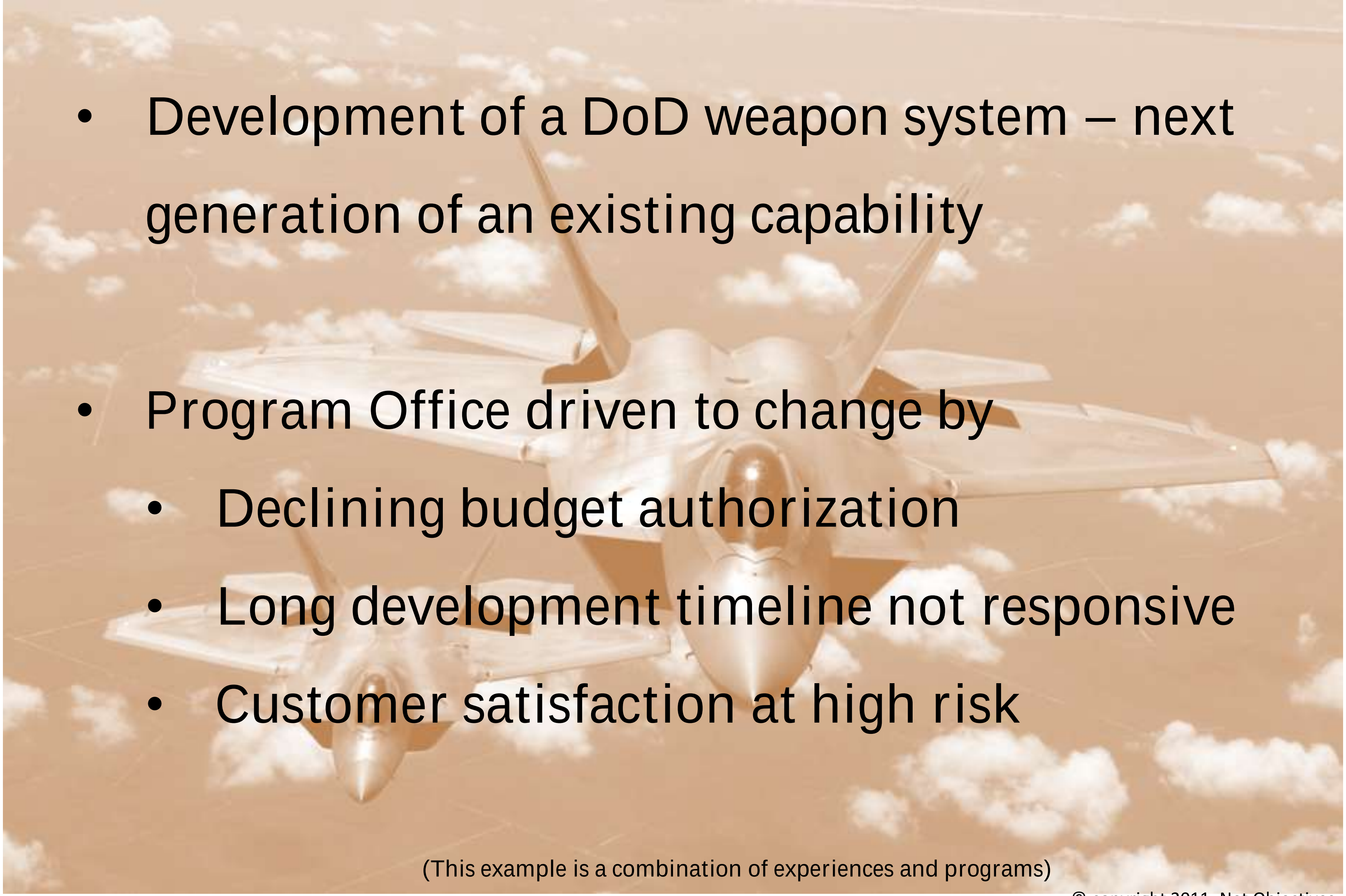
Only User Stories have “Business Value”! (sorry devs)

Transition Thinking: Big batch to smaller continuous incremental batches:

PO: highest business value, right size *at the right time* (just in time)

Requires continuous planning

Case Study – DoD Acquisition

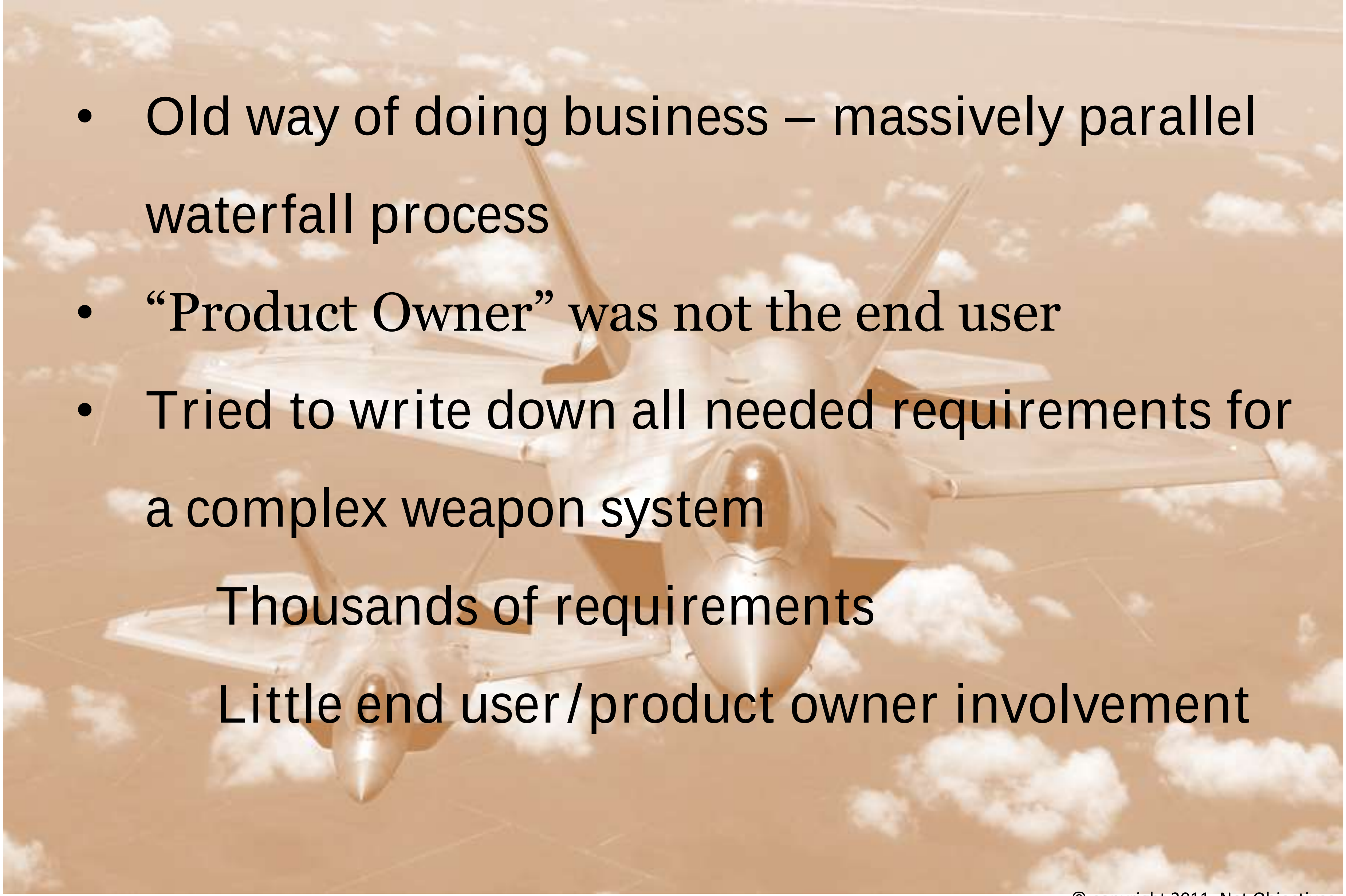


- Development of a DoD weapon system – next generation of an existing capability

- Program Office driven to change by
  - Declining budget authorization
  - Long development timeline not responsive
  - Customer satisfaction at high risk

(This example is a combination of experiences and programs)

How did the Product Owner Act?



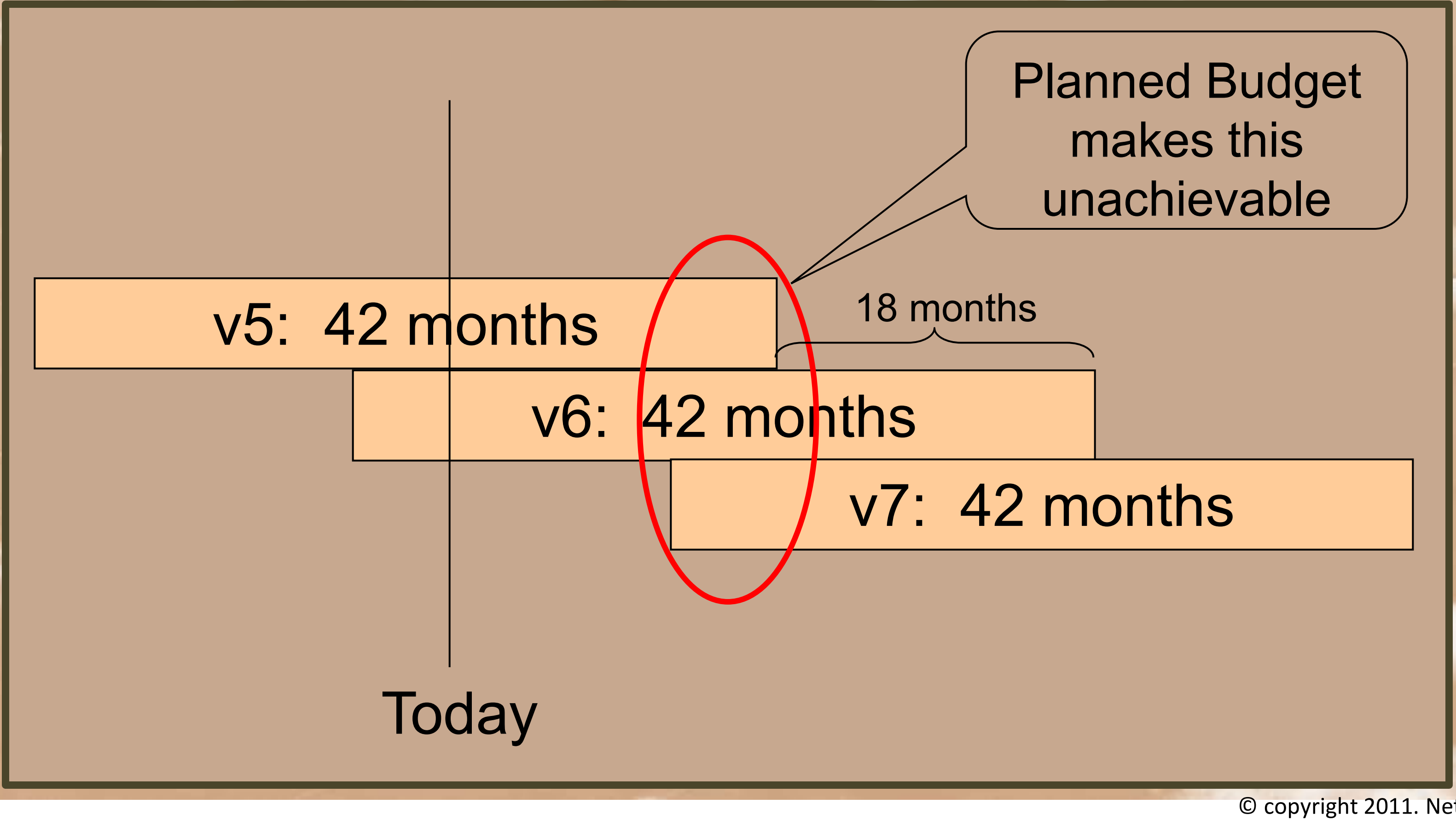
- Old way of doing business – massively parallel waterfall process
- “Product Owner” was not the end user
- Tried to write down all needed requirements for a complex weapon system

  Thousands of requirements

  Little end user/product owner involvement

How did the Product Owner Act?

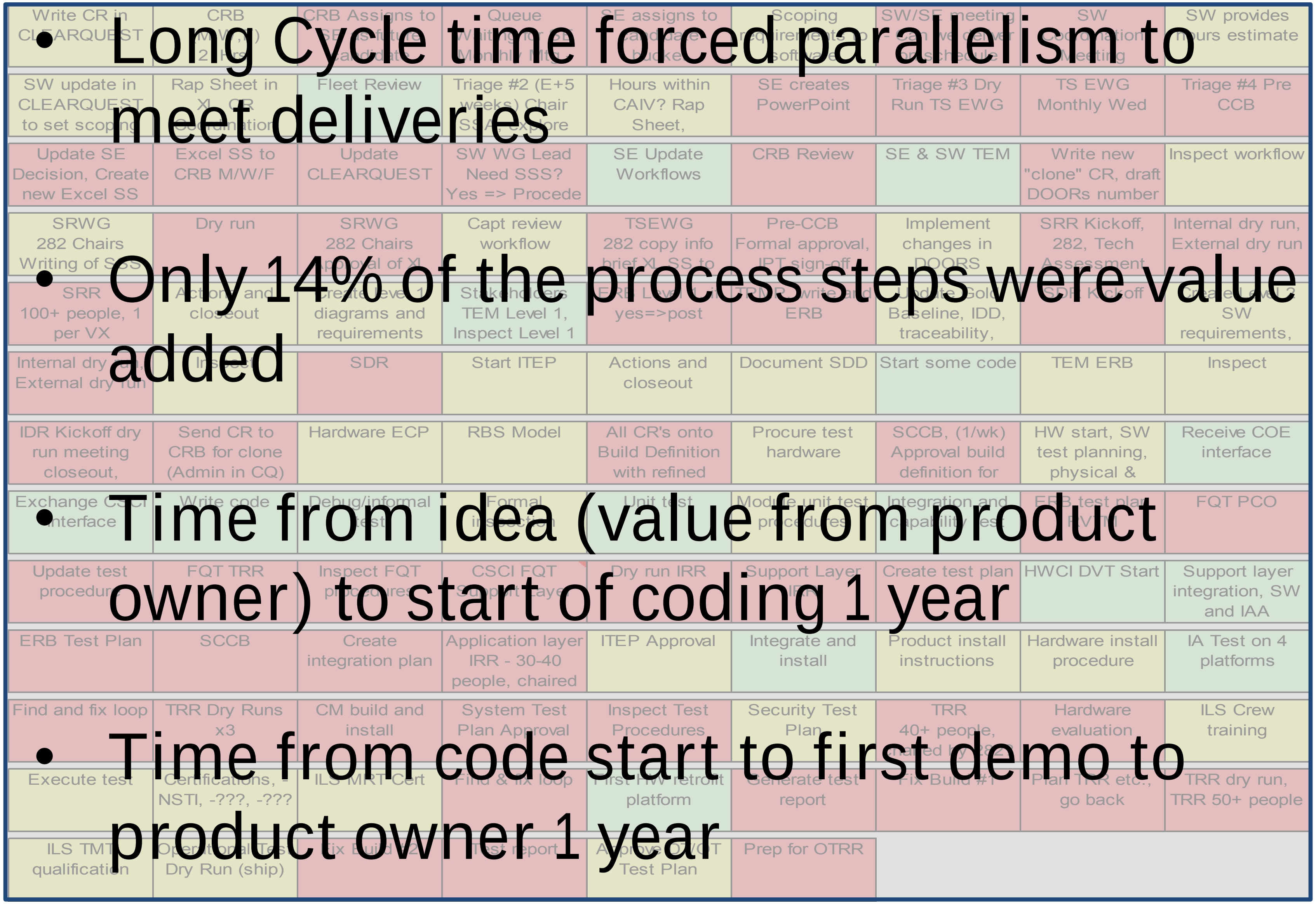
- Old way of doing business – massively parallel waterfall process



Planned Budget makes this unachievable

Case Study – the Old Way

- Long Cycle time forced parallelism to meet deliveries
- Only 14% of the process steps were value added
- Time from idea (value from product owner) to start of coding 1 year
- Time from code start to first demo to product owner 1 year



Write CR in CLEARQUEST	CRB (M,W,F) 2 Hrs	CRB Assigns to SE as future candidate	Queue Waiting for SE Monthly Mtg	SE assigns to candidate bucket	Scoping requirements to software	SW/SE meeting - can we deliver on schedule	SW coordination Meeting	SW provides hours estimate
SW update in CLEARQUEST to set scoping	Rap Sheet in XL, CR coordination	Fleet Review	Triage #2 (E+5 weeks) Chair SSA, explore	Hours within CAIV? Rap Sheet,	SE creates PowerPoint	Triage #3 Dry Run TS EWG	TS EWG Monthly Wed	Triage #4 Pre CCB
Update SE Decision, Create new Excel SS	Excel SS to CRB M/W/F	Update CLEARQUEST	SW WG Lead Need SSS? Yes => Procede	SE Update Workflows	CRB Review	SE & SW TEM	Write new "clone" CR, draft DOORs number	Inspect workflow
SRWG 282 Chairs Writing of SSS	Dry run	SRWG 282 Chairs approval of XL	Capt review workflow	TSEWG 282 copy info brief XL SS to	Pre-CCB Formal approval, IPT sign-off	Implement changes in DOORS	SRR Kickoff, 282, Tech Assessment	Internal dry run, External dry run
SRR 100+ people, 1 per VX	Actions and closeout	Create Level 1 diagrams and requirements	Stakeholders TEM Level 1, Inspect Level 1	ERB Level 1 in yes=>post	TRMP, write and ERB	Update Gold Baseline, IDD, traceability,	SDR Kickoff	Create Level 2 SW requirements,
Internal dry run, External dry run	Inspect	SDR	Start ITEP	Actions and closeout	Document SDD	Start some code	TEM ERB	Inspect
IDR Kickoff dry run meeting closeout,	Send CR to CRB for clone (Admin in CQ)	Hardware ECP	RBS Model	All CR's onto Build Definition with refined	Procure test hardware	SCCB, (1/wk) Approval build definition for	HW start, SW test planning, physical &	Receive COE interface
Exchange CSCI interface	Write code	Debug/informal test	Formal inspection	Unit test	Module unit test procedures	Integration and capability test	ERB test plan RVTM	FQT PCO
Update test procedure	FQT TRR	Inspect FQT procedures	CSCI FQT Support Layer	Dry run IRR	Support Layer IRR	Create test plan	HWCI DVT Start	Support layer integration, SW and IAA
ERB Test Plan	SCCB	Create integration plan	Application layer IRR - 30-40 people, chaired	ITEP Approval	Integrate and install	Product install instructions	Hardware install procedure	IA Test on 4 platforms
Find and fix loop	TRR Dry Runs x3	CM build and install	System Test Plan Approval	Inspect Test Procedures	Security Test Plan	TRR 40+ people, chaired by 282	Hardware evaluation	ILS Crew training
Execute test	Certifications, - NSTI, -???, -???	ILS MRT Cert	Find & fix loop	First HW retrofit platform	Generate test report	Fix Build #1	Plan TRR etc., go back	TRR dry run, TRR 50+ people
ILS TMT qualification	Operational Test Dry Run (ship)	Fix Build #2	Test report	Approve OVOT Test Plan	Prep for OTRR			

Case Study – the Old Way

- The delivered system was not acceptable to the end user

- New requirements – evolved after contract award – could not be met at all

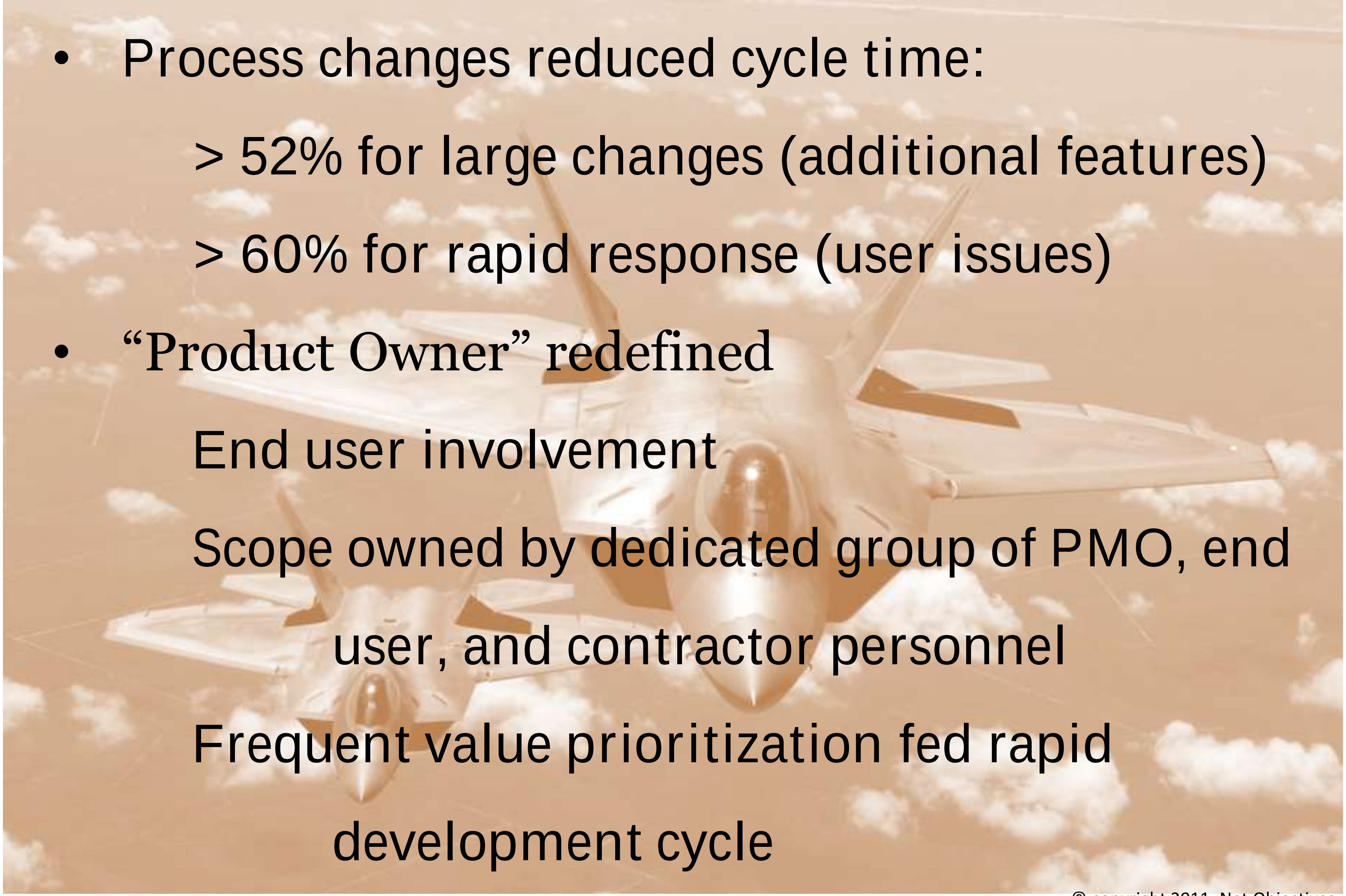
- Real product owner involvement was lacking in the process – and it showed in the result!

Agile Development

The process was changed by applying lean/agile to the system development – required a new definition and role for the product owner!

Net Objectives

Case Study - Results



- Process changes reduced cycle time:

  > 52% for large changes (additional features)

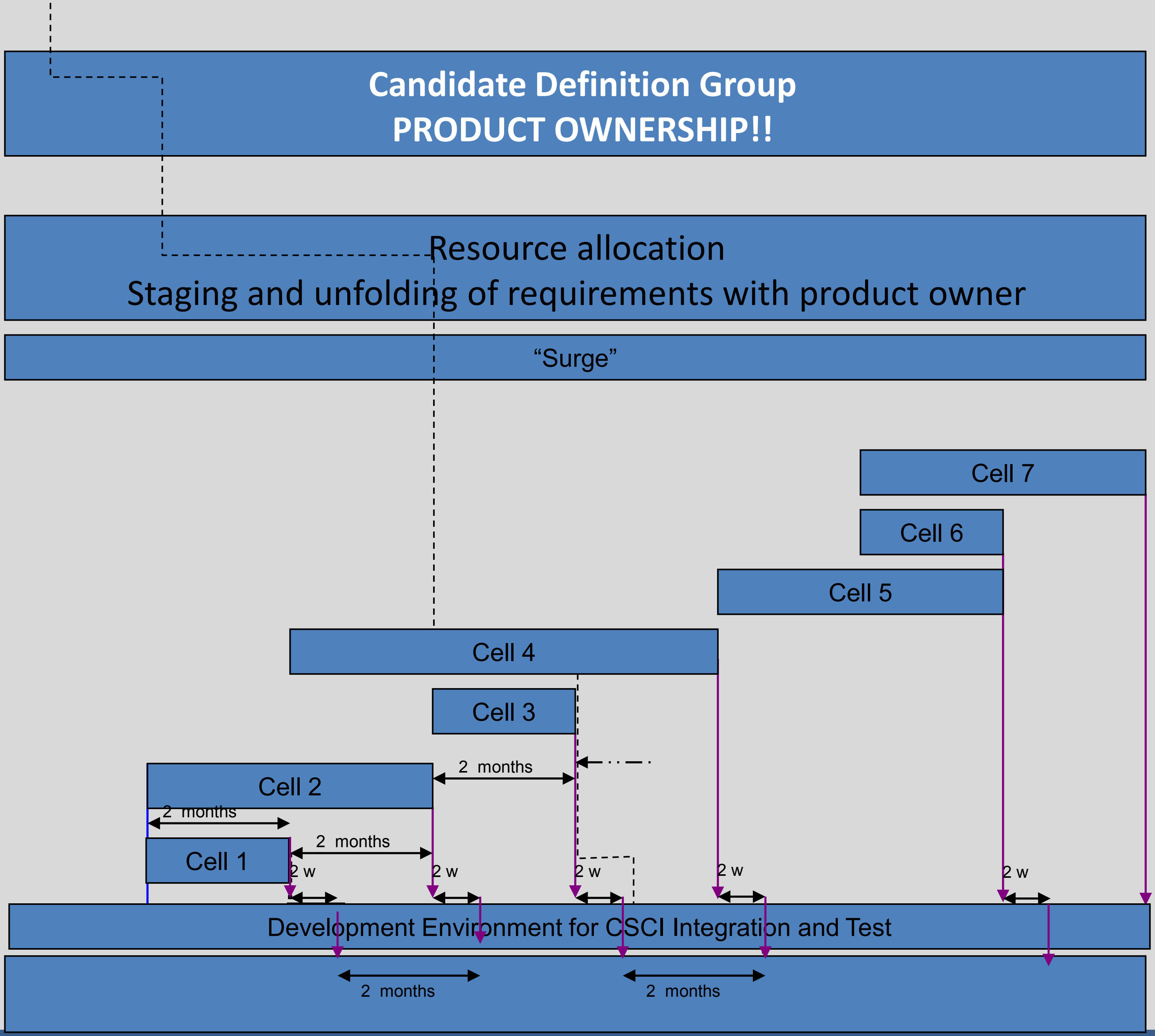
  > 60% for rapid response (user issues)

- “Product Owner” redefined

  End user involvement

  Scope owned by dedicated group of PMO, end user, and contractor personnel

  Frequent value prioritization fed rapid development cycle



Candidate Definition Group

PRODUCT OWNERSHIP!!

Resource allocation

Staging and unfolding of requirements with product owner

“Surge”

Cell 7

Cell 6

Cell 5

Cell 4

Cell 3

Cell 2

Cell 1

2 months

2 months

2 months

2 w

2 w

2 w

2 w

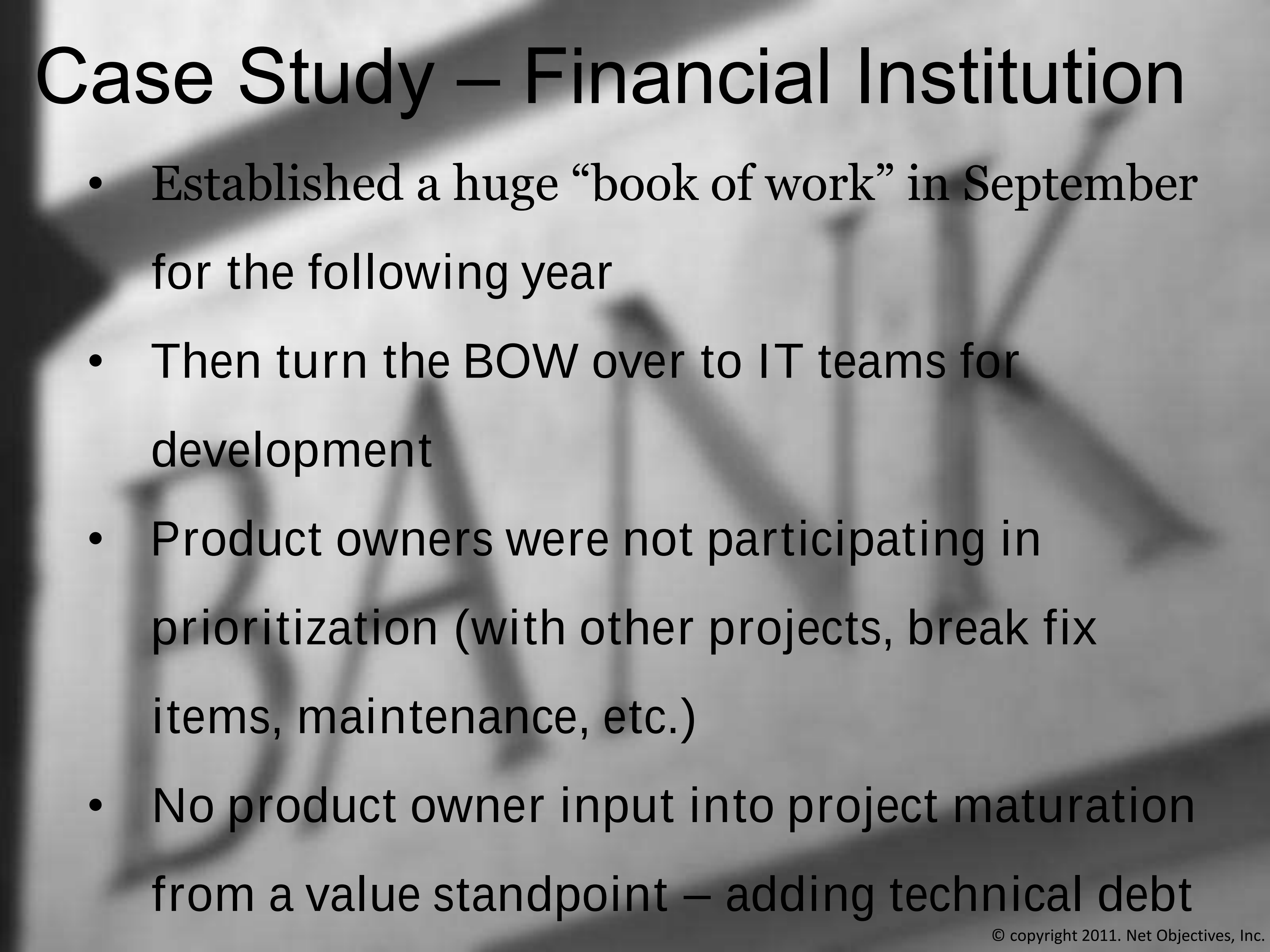
2 w

Development Environment for CSCI Integration and Test

2 months

2 months

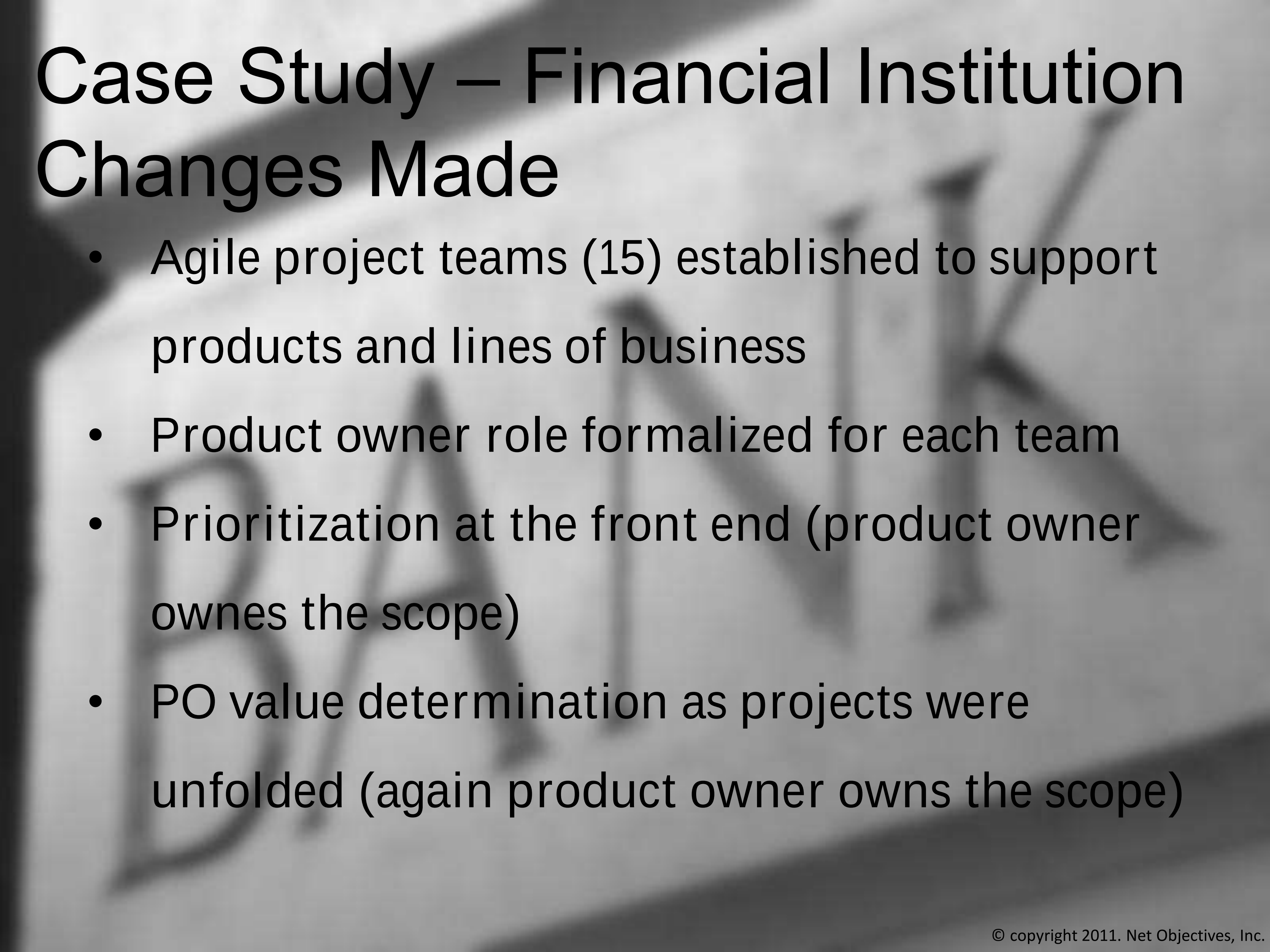
Case Study – Financial Institution



- Established a huge “book of work” in September for the following year
- Then turn the BOW over to IT teams for development
- Product owners were not participating in prioritization (with other projects, break fix items, maintenance, etc.)
- No product owner input into project maturation from a value standpoint – adding technical debt

"You cannot build the right thing if you have not discovered it first!"

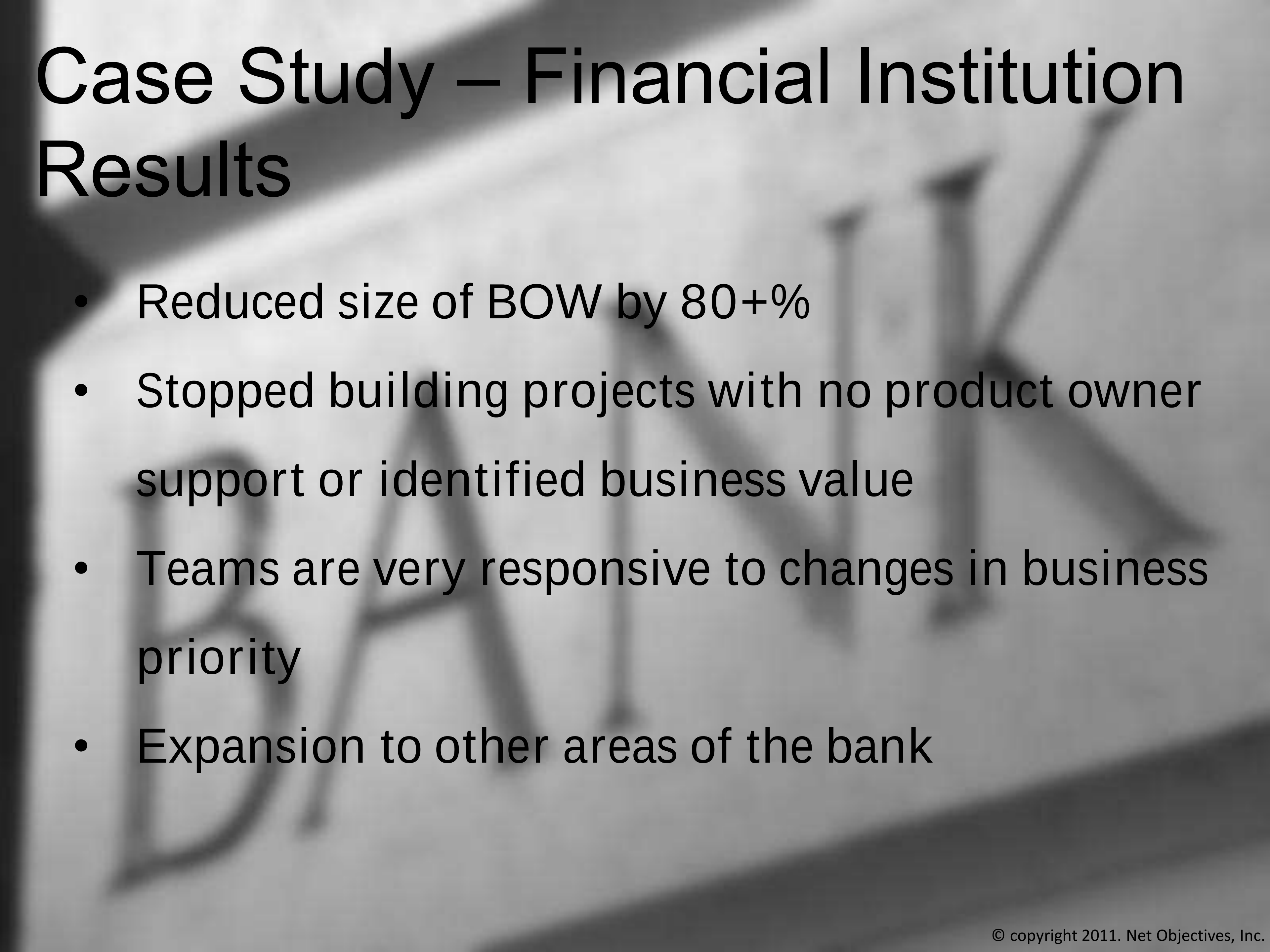
This is the role of the product owner in agile development!



Case Study – Financial Institution Changes Made

- Agile project teams (15) established to support products and lines of business
- Product owner role formalized for each team
- Prioritization at the front end (product owner ownes the scope)
- PO value determination as projects were unfolded (again product owner owns the scope)

Case Study – Financial Institution Results



- Reduced size of BOW by 80+%
- Stopped building projects with no product owner support or identified business value
- Teams are very responsive to changes in business priority
- Expansion to other areas of the bank

Questions

?

Michael Cox
Vice President and Senior Consultant
NetObjectives, Inc.
michael.cox@netobjectives.com
610-858-7289

Secure Agile Development

Jeffery Payne

jeff.payne@coveros.com

Chief Executive Officer

Contents

- About Coveros
- SecureAgile development process
- Integrating security into Agile development
- Q&A

Who we are

- Coveros helps organizations accelerate the delivery of secure, reliable software

- SecureAgile Services
  - Secure software development services
  - Application vulnerability remediation
  - Application security assessments and testing
  - Agile software process improvement
- SecureCI Product
  - Open source secure continuous integration product

- Our primary markets
  - Defense systems
  - National security
  - Healthcare
  - Financial services

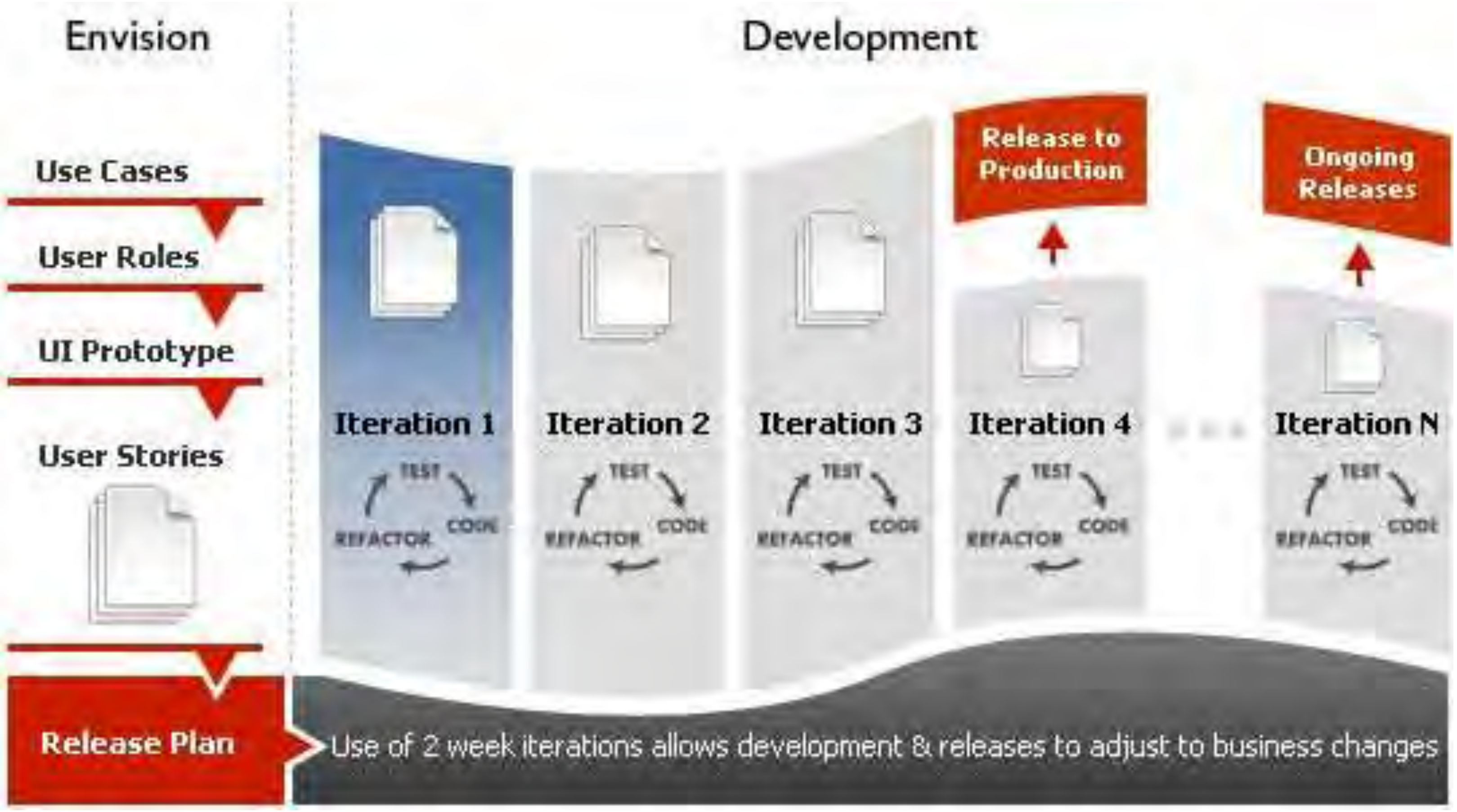
Corporate Partners

ARXAN

FORTIFY SOFTWARE

SQE TRAINING

SecureAgile^{TM} Development Process



Assures time-to-market while achieving security objectives

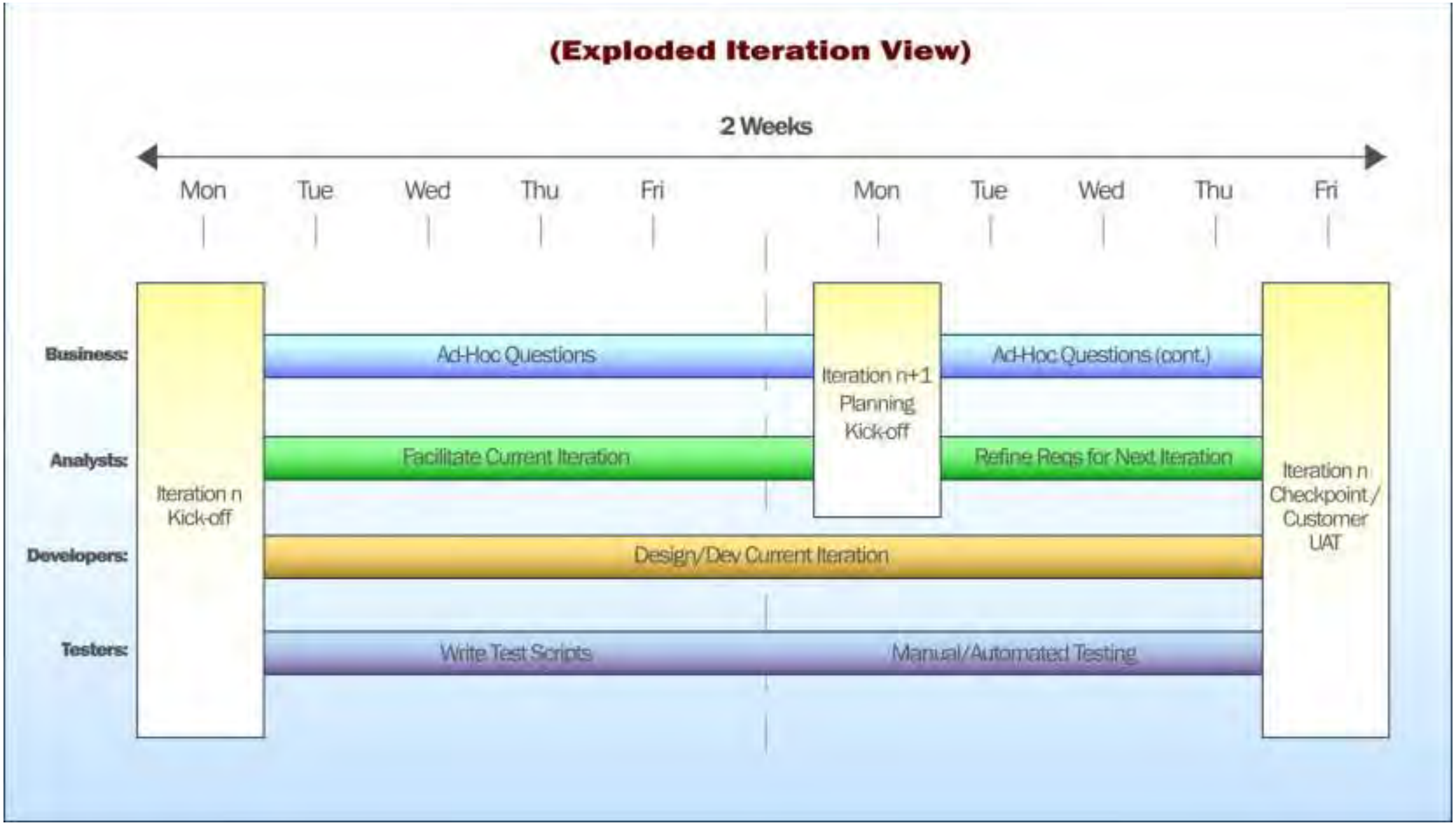
Envision Process (aka Initial Planning)

- Create User Personas to keep the customer top of mind
- Develop Use Cases to understand overall business process
- Build Global Backlog of User Stories with priority
- Prototype UI as appropriate / necessary
- Define initial application architecture and address initial research spikes
- Develop Release Plan comprised of Stories within Iterations
- Create test strategy / master test plan for project

Security Activities within Envision

- Threat modeling / Architectural Risk Analysis to understand threats, possible attacks, and value of assets
- Misuse / Abuse Case development
- Incorporate security requirements into User Stories
  - “User will not” nomenclature as needed
- Develop high level security test strategy / plan
- Understand compliance & regulatory needs

Iterative Development Process


(Exploded Iteration View)

2 Weeks

Mon, Tue, Wed, Thu, Fri, Mon, Tue, Wed, Thu, Fri

Business: Ad-Hoc Questions; Ad-Hoc Questions (cont.)

Analysts: Facilitate Current Iteration; Refine Reqs for Next Iteration

Developers: Design/Dev Current Iteration

Testers: Write Test Scripts; Manual/Automated Testing

Iteration n Kick-off

Iteration n+1 Planning Kick-off

Iteration n Checkpoint / Customer UAT

Defensive design and coding

- Incorporation of security controls into software design and code
  - Security frameworks like OWASP ESAPI

- Use of vetted components
  - Libraries of secure components

- Examination of design / code looking for realization of architectural risks

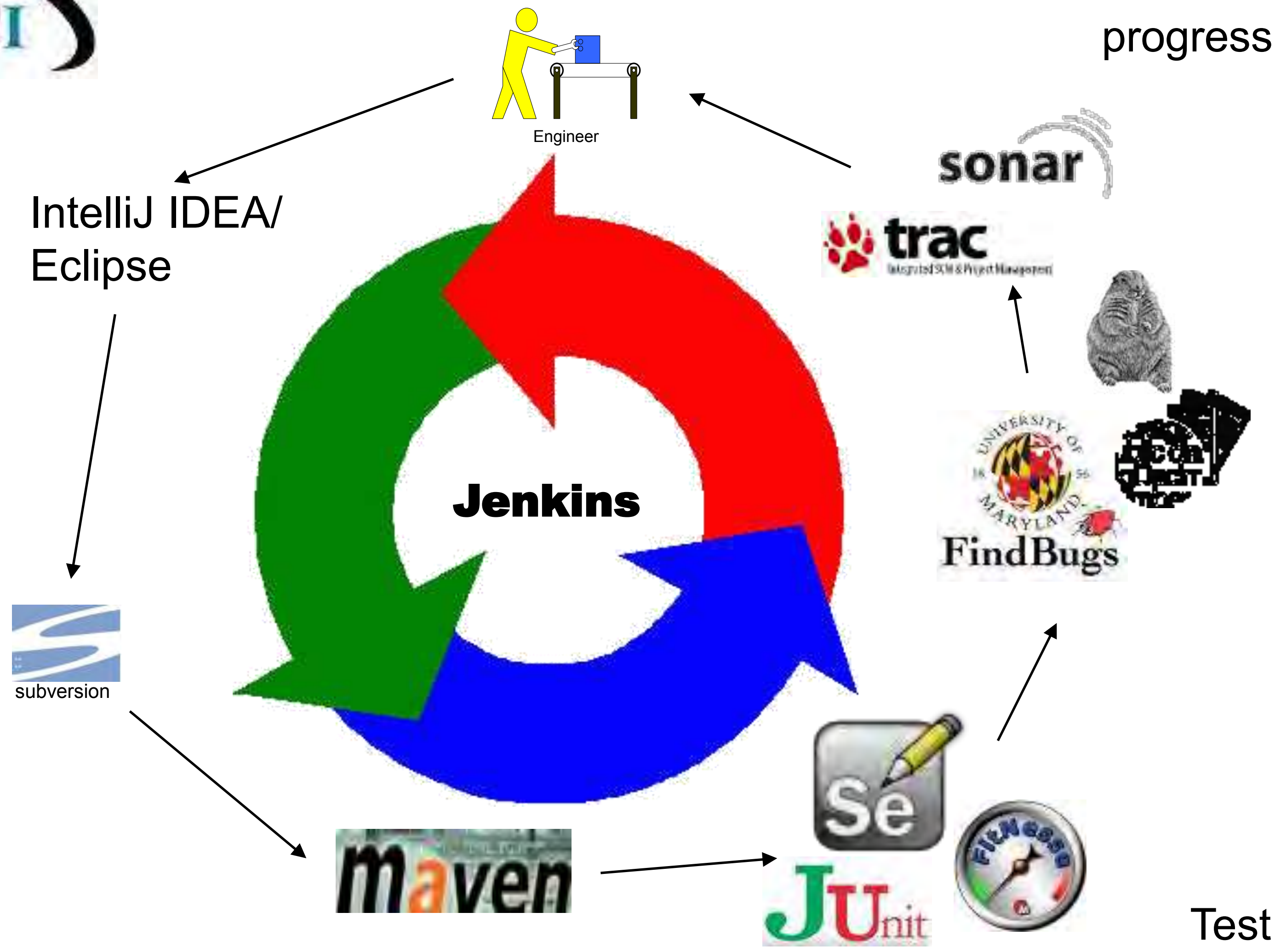
Software Assurance

- Secure code review
  - Both automated and manual
- Security testing
  - Risk-based testing
  - Testing of security functionality
- Penetration testing

Continuous Integration

- Automation of build, test, deploy process
  - Check-in builds / tests
  - Nightly code integrations and regression tests
  - Automated promotion between test stages
  - Automated notification of build failures
- A critical capability to have when building software using agile
- Many good open source products available

SecureCI



Track progress

Create code

IntelliJ IDEA/ Eclipse

Version code

Build application

Test application

Test security

Questions?

Thank You

The Agile Process

Jeffery Payne
CEO, Coveros, Inc.
jeff.payne@coveros.com

Agenda

- Introductions & Expectations
- What is Agile?
- Why does Agile work?
- Myths about Agile
- Agile development process
- Wrap Up

About Coveros

coveros

- Coveros helps organizations accelerate the delivery of secure, reliable software

- Our consulting services:
  - Agile software development
  - Application security
  - Software quality assurance
  - Software process improvement

- Our key markets:
  - Financial services
  - Healthcare
  - Defense
  - National security

Corporate Partners

ARXAN

FORTIFY SOFTWARE

Telos

SQE TRAINING

Introductions

Instructor – Jeffery Payne

Jeffery Payne is CEO and founder of Coveros, Inc., a software company that helps organizations accelerate the delivery of secure, reliable software. Coveros uses agile development methods and a proven software assurance framework to build security and quality into software from the ground up. Prior to founding Coveros, Jeffery was Chairman of the Board, CEO, and co-founder of Cigital, Inc. Under his direction, Cigital became a leader in software security and software quality solutions, helping clients mitigate the risk of software failure. Jeffery is a recognized software expert and popular speaker at both business and technology conferences on a variety of software quality, security, and agile development topics. He has also testified before Congress on issues of national importance, including intellectual property rights, cyber-terrorism, Software research funding, and software quality.

Class Attendees

Expectations

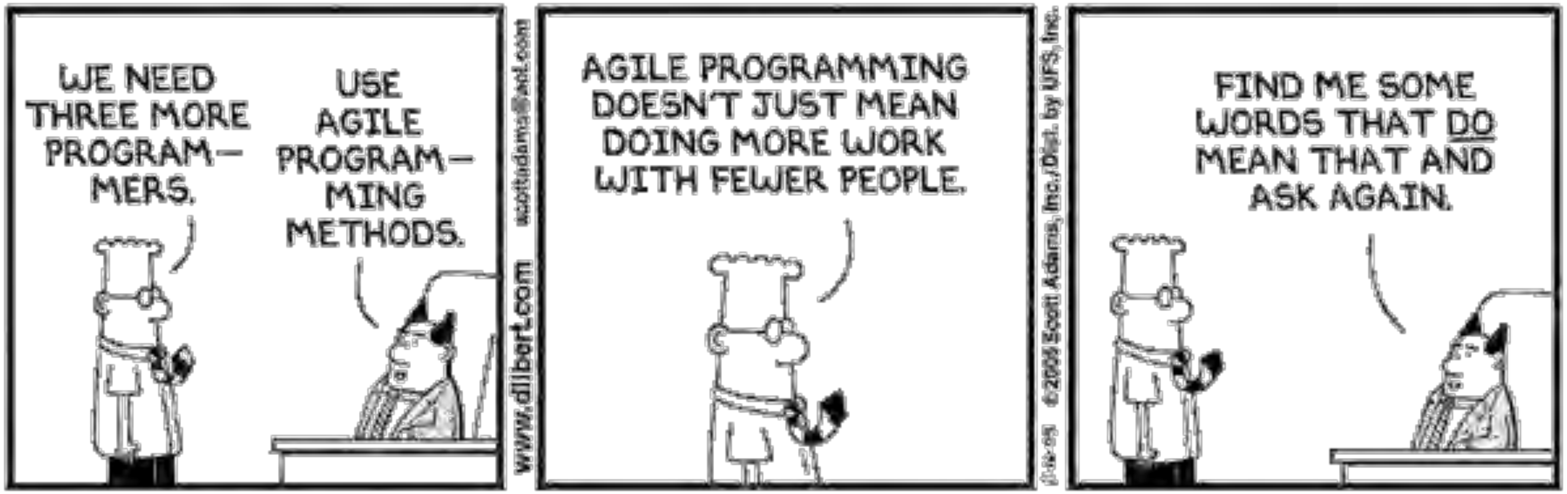
- What are your expectations for this class?
- What do you wish to learn?
- What questions do you want answered?

Objectives

The primary objectives of this course are to:

- Introduce you to Agile software development
- Discuss the major differences between Agile and traditional methodologies.
- Describe how Agile practices and principles improve the software development process.

What is Agile?

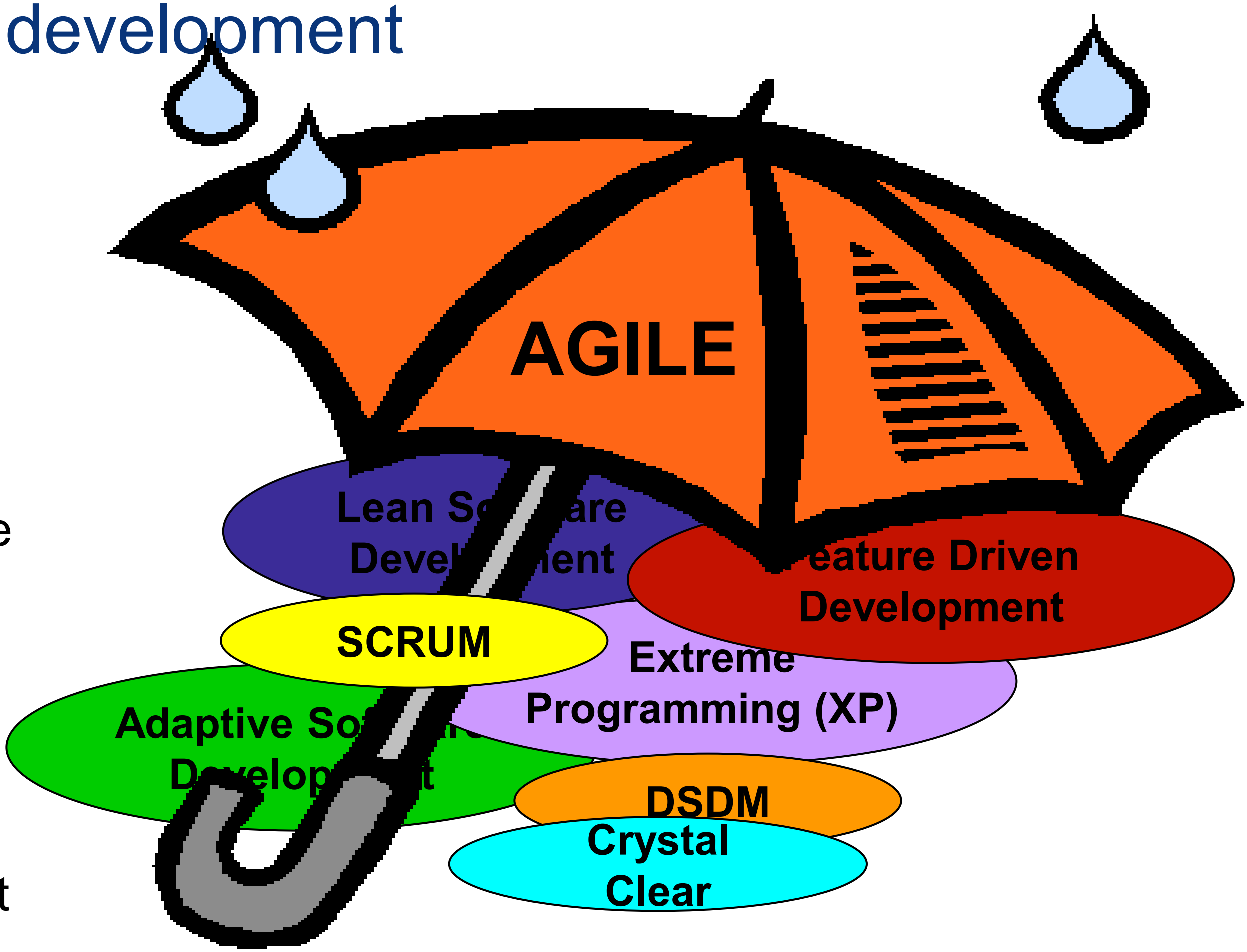

WE NEED THREE MORE PROGRAM-MERS.
USE AGILE PROGRAM-MING METHODS.
scottadams@aol.com
www.dilbert.com
AGILE PROGRAMMING DOESN'T JUST MEAN DOING MORE WORK WITH FEWER PEOPLE.
© 2003 Scott Adams, Inc./Dist. by UFS, Inc.
FIND ME SOME WORDS THAT DO MEAN THAT AND ASK AGAIN.

© Scott Adams, Inc./Dist. by UFS, Inc.

What is Agile?

The agile movement began as a set of ideas for improving software development

- Close collaboration between programmers & business people
- Face-to-face communication
- Frequent delivery of deployable business value
- Self-organizing teams
- Crafting code & environment to support requirements changes
- The most important output of a project is working software


AGILE
Lean Software Development
Feature Driven Development
SCRUM
Extreme Programming (XP)
Adaptive Software Development
DSDM
Crystal Clear

http://www.agilemanifesto.org

What is Agile?

Different agile methodologies emphasize different practices

Scrum

- Product Backlog
- Sprint Backlog
- Daily Scrum
- Sprint Review
- Self-Directed Teams
- Chickens and Pigs

DSDM

- Timeboxing
- Meta Modeling
- MoSCoW Method

XP

- Test-Driven Development
- Refactoring
- Simple Design
- Pair Programming
- Collective Ownership
- Coding Standard
- Sustainable Pace
- Metaphor
- Continuous Integration
- The Planning Game
- Small Releases
- On-Site Customer

Lean

- Seeing waste
- Value stream mapping
- Set-based development
- Pull systems
- Queuing theory
- Motivation
- Measurements

Crystal

- Reflective Improvement
- Osmotic Communication
- Easy Access to Expert Users

What is Agile?

All agile methodologies adhere to some basic principles

- Early and continuous delivery of valuable software
- Welcome changing requirements, even late in development
- Deliver working software frequently
- Business people and developers work together daily
- Build projects around motivated individuals and trust them to get the job done.
- Frequent conversation to convey information efficiently
- Working software as the primary measure of progress
- Sustainable development
- Continuous attention to technical excellence and good design
- Simplicity—maximizing the amount of work not done
- The best architectures, requirements, and designs emerge from self-organizing teams
- At regular intervals, the team reflects on, tunes, and adjusts its behavior

What is Agile?

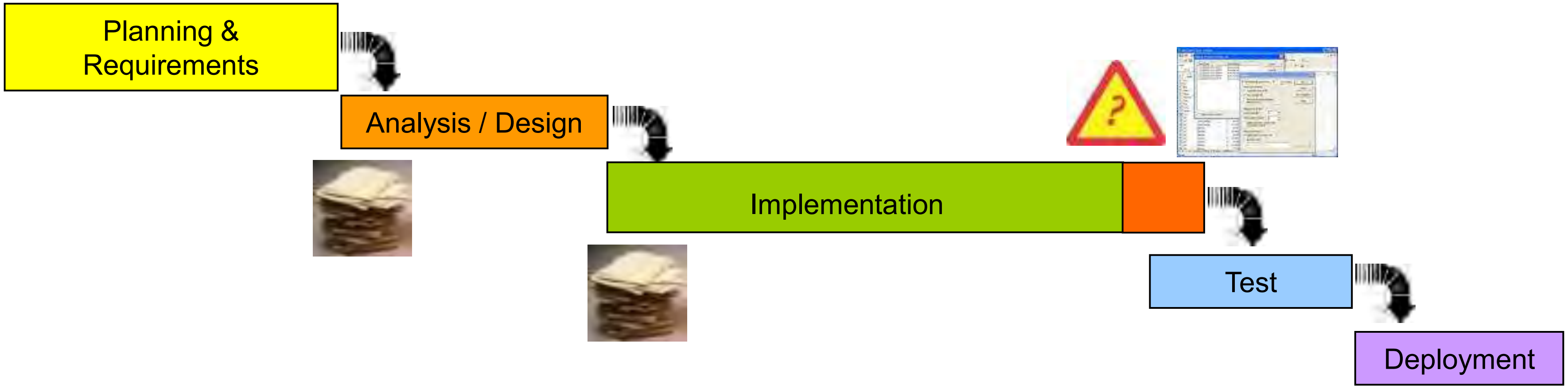
The agile approach is a different way of *thinking* about SW projects

Phase-Based	Agile
Plan Driven	Learning Driven
Infrequent Client Communication	Continuous Client Communication
Deliver Once in "Big Bang" Fashion, Typically 9 – 12 Months	Deliver in Short, Business-Focused Releases, Typically 2 – 3 Months
Develop in Distinct Phases with Interim Paper Deliverables	Develop in 2-Week Long Sprints and Deliver Working Code
Develop in Layers: Presentation, Persistence, Business, etc.	Develop in End-to-End Functional Slices
View Programming as Construction	View Programming as Design
Integration of Different Layers Occurs at End of Build Phase	Continuously Integrate Code Throughout (Hourly Builds)
Testing as Separate Phase at End of Project, Typically Emphasizing Functional Level	Fully-Automated, Continuous Testing at Both Functional and Unit Level
High Cost of Change	Low Cost of Change

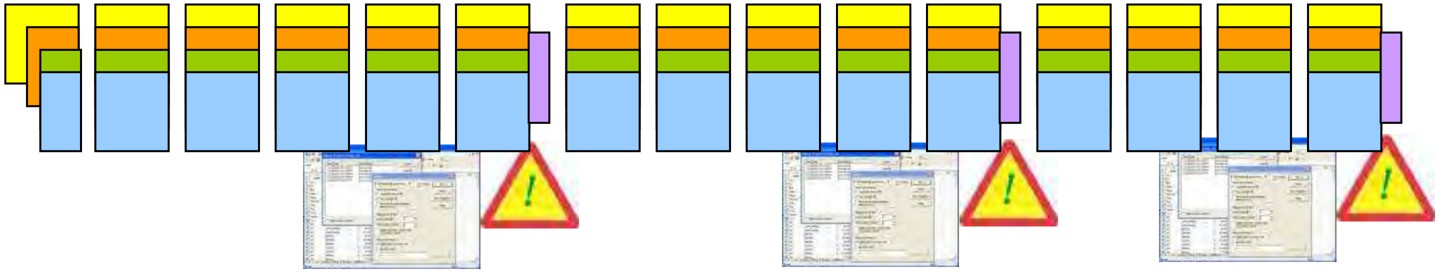
What is Agile?

Agile Rearranges Key Development Activities

Phase-Based approach

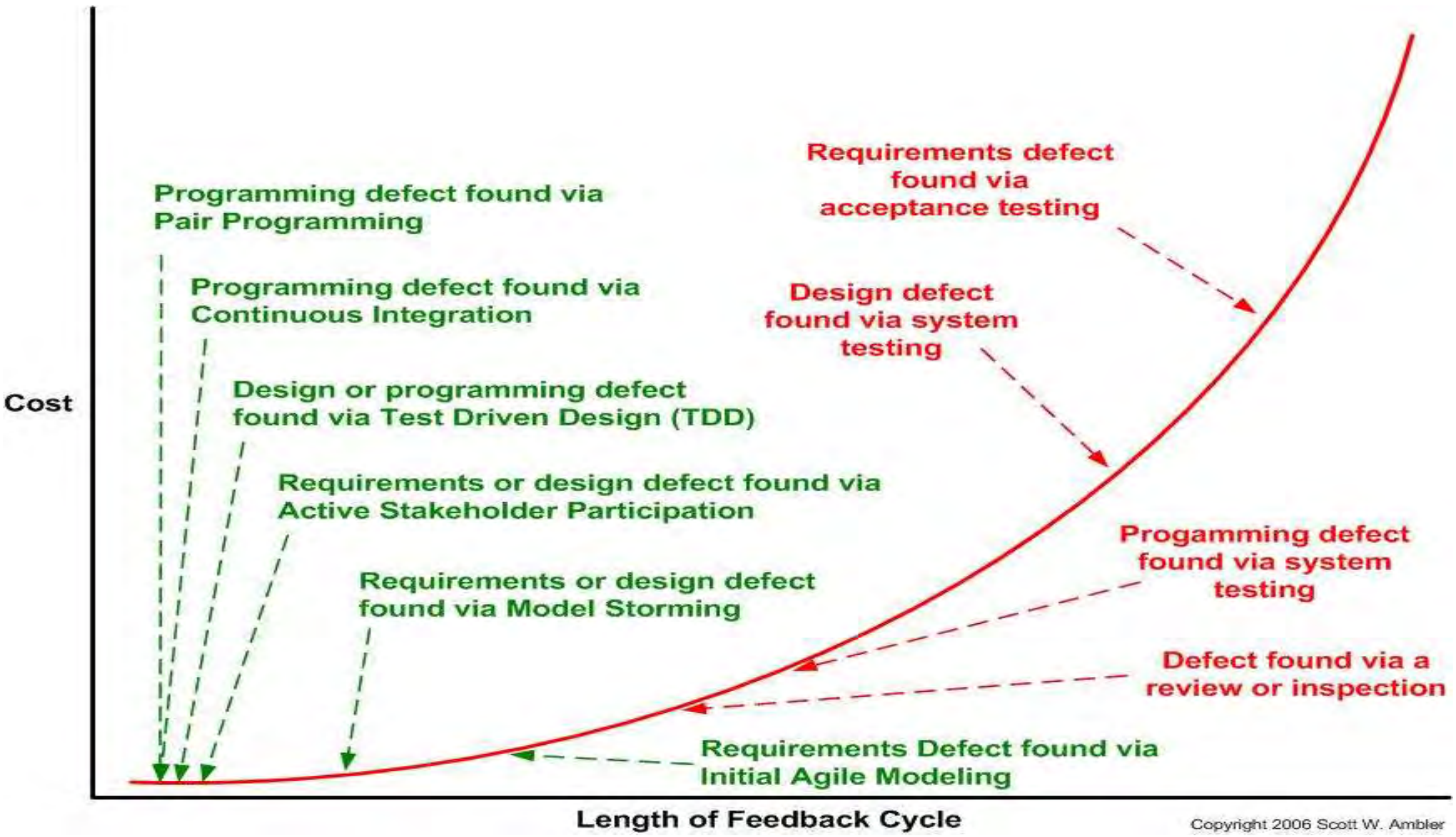


Incremental/Agile approach



Note: Agile is not simply a set of —smallwaterfalls”

Why Projects Fail: Cost of change

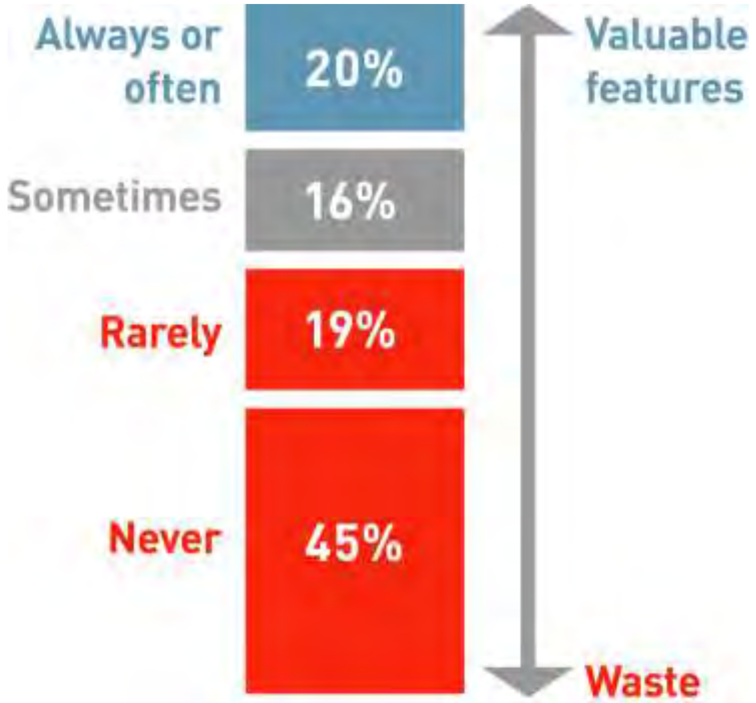


Why Agile?

Why Projects Fail: Poor requirements management

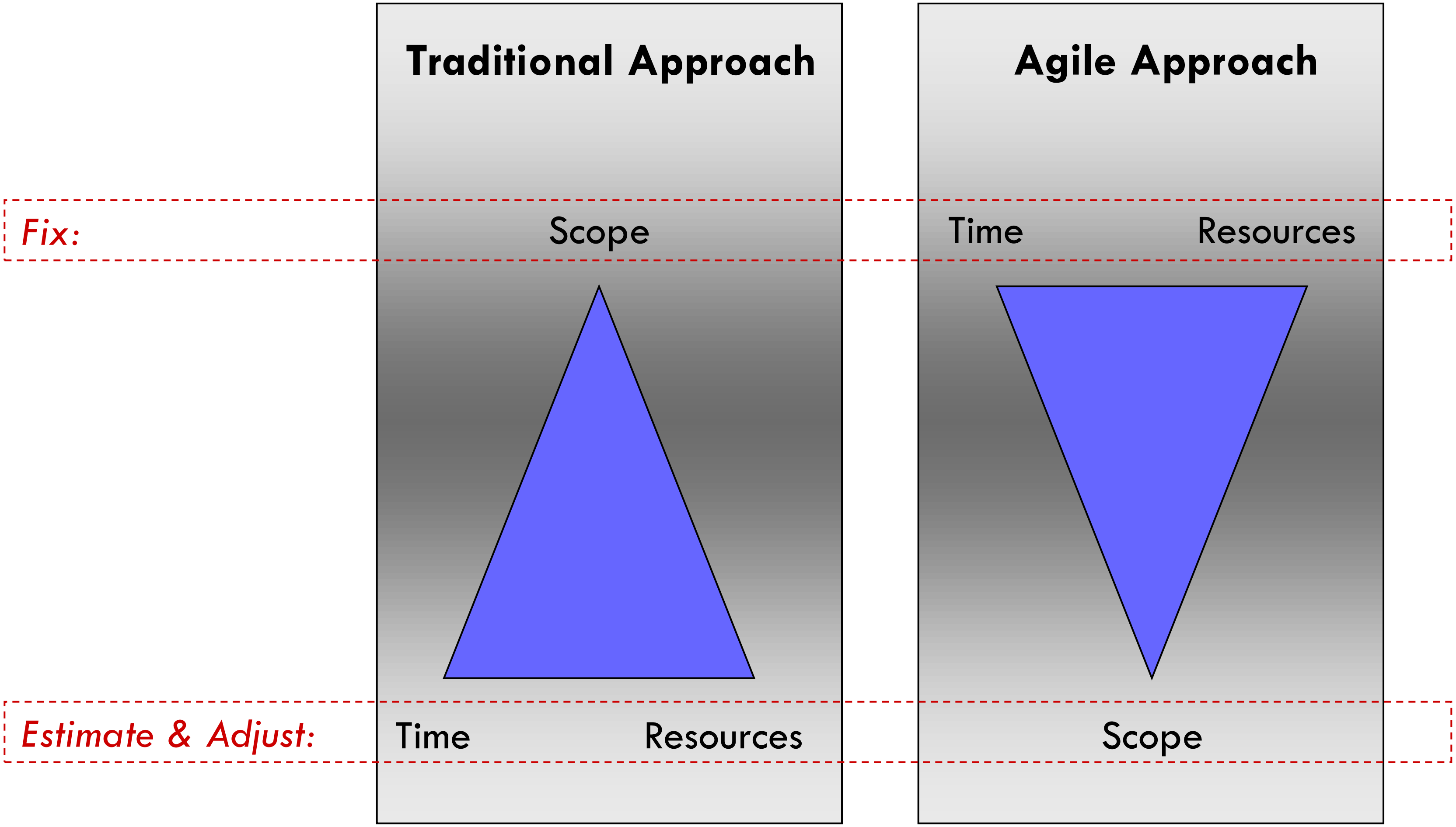
Much of present-day software acquisition procedures rests upon the assumption that one can specify a satisfactory system in advance. . . . I think **this assumption is fundamentally wrong,** and that many software acquisition problems spring from that fallacy. --Fred Brooks, 1986

Standish Group study on "Features and Functions used in a Typical System"



Why Agile?

Addressing these Issues with Agile Planning



	Traditional Approach	Agile Approach
Fix:	Scope	Time Resources
Estimate & Adjust:	Time Resources	Scope

Why Agile?

Addressing these Issues with Agile Software Delivery

- Deliver incremental change in order to maximize feedback.
- Accept change continuously in order to minimize waste.

THE AGILE BET

If the cost of change can be kept low over time, the cost savings that result from **early feedback will far** outweigh the added costs of early change.

Myths about Agile

- There are many myths floating around about Agile Development

- These myths are often due to:
  - A lack of understanding of Agile
  - Early thinking within the Agile community that proved to be wrong
  - Trying to implement Agile in a manner that will not work
  - Relying upon consultants who know the theory but can‘t apply it pragmatically

- Regardless, Agile is not a silver bullet that will magically transform your organization without a lot of hard work

Agile Myths

No planning takes place on Agile projects

- Reality:
  - Agile teams spend as much, if not more, time planning development activities.
  - The major difference is that the planning is spread throughout the entire lifecycle of the project.
  - Traditional methodologies emphasize lots of upfront planning. Agile teams do some planning upfront, but only enough to understand the major milestones and dependencies.
  - Agile is designed to embrace change and uncertainty, so most planning is done in a continuous, _just-in-time' fashion.

- Planning Pragmatics
  - Define your wish list / vision up front
  - Define an initial, high level architecture up front
  - Wireframe your user interface look and feel
  - Detail out 1-2 Sprints of work
  - Detail out additional Sprints prior to the start

Agile Myths

No documentation is written on Agile projects

- Reality:
  - Agile emphasizes working software over comprehensive documentation
  - Agile encourages the —right” amount of documentation, that is, documents that are of value to the project and downstream maintenance
  - The creation of a document is treated as a requirement, which in turn must be estimated. This forces the team to carefully consider the costs of documentation and focus only on the development of concise, valuable documentation.

- Documentation Pragmatics
  - Document as necessary for communications
  - Document as necessary for support and maintenance
  - Document as necessary for corporate policy and/or regulatory compliance

Agile Myths

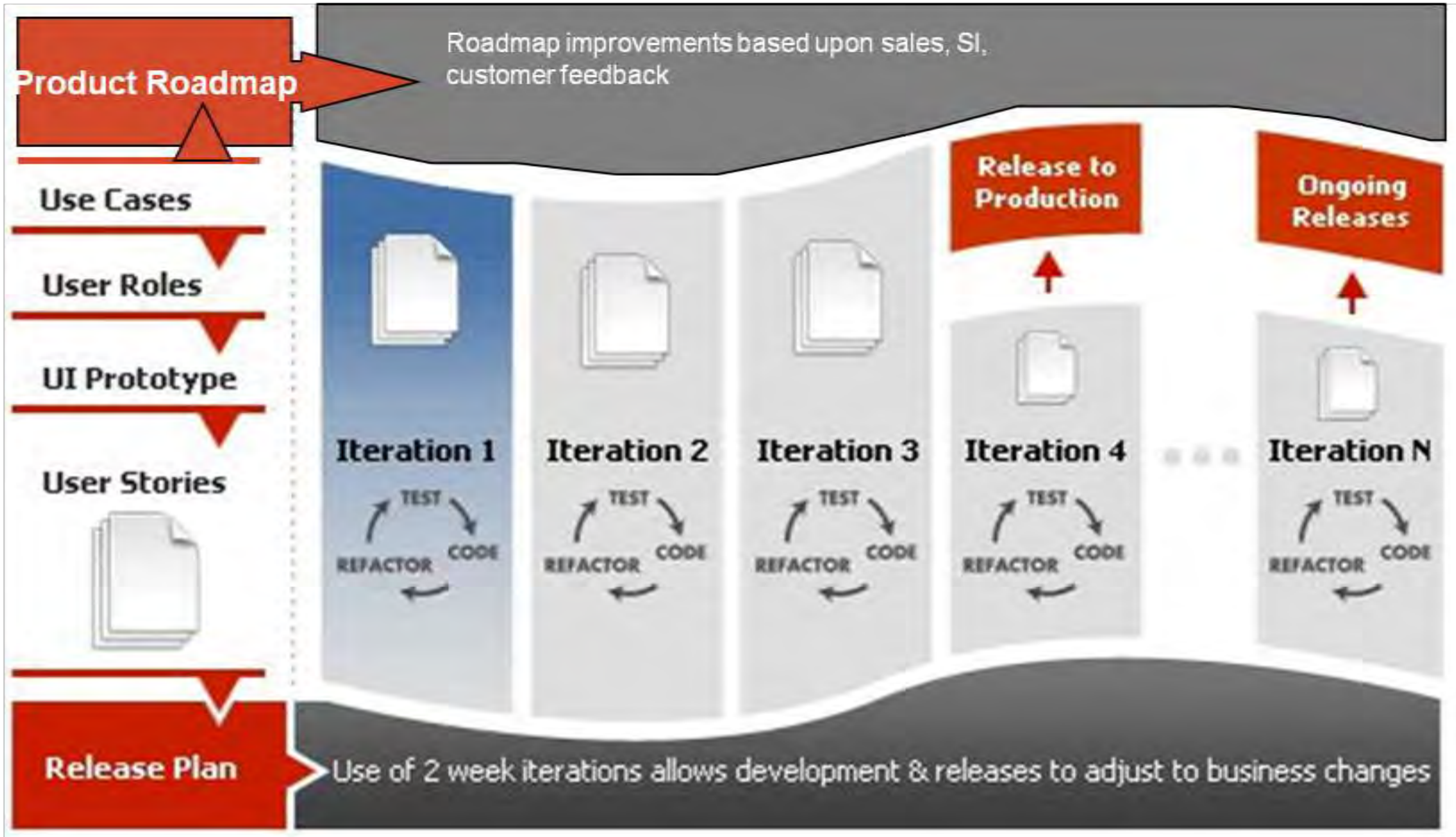
Software testers aren‘t needed on Agile projects

- Reality:
  - In Agile development, just like in traditional methods, the development and test team share responsibility for code quality.
  - More frequent code deployment to the test environment requires enhanced methods to ensure quality, such as test automation and functional test suites.
  - These activities require a skilled and capable test team to execute successfully.
  - Agile does rely on more automated testing and testing inline with development vs. post development so testers do test in a different way
- Testing Pragmatics
  - Unit testing by development is a necessity
  - A test automation strategy should be used to dictate where the ROI point is for automation
  - Automated testing is integrated with continuous integration to support rapid build, test, fix cycles
  - Full integration and system testing is done mostly prior to release

Agile Product Development Process

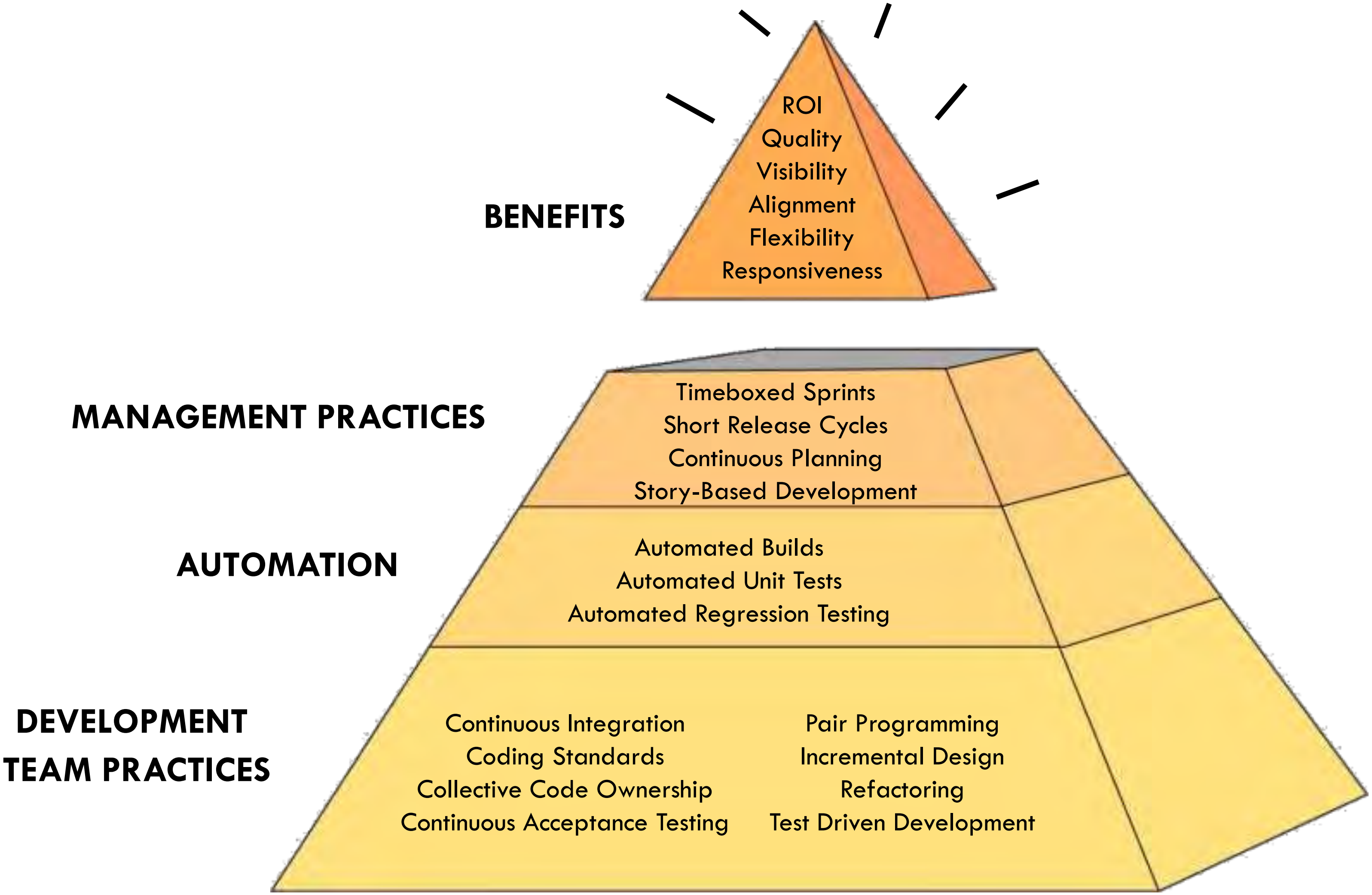
Initial Planning

On-going Planning, Implementation & Release

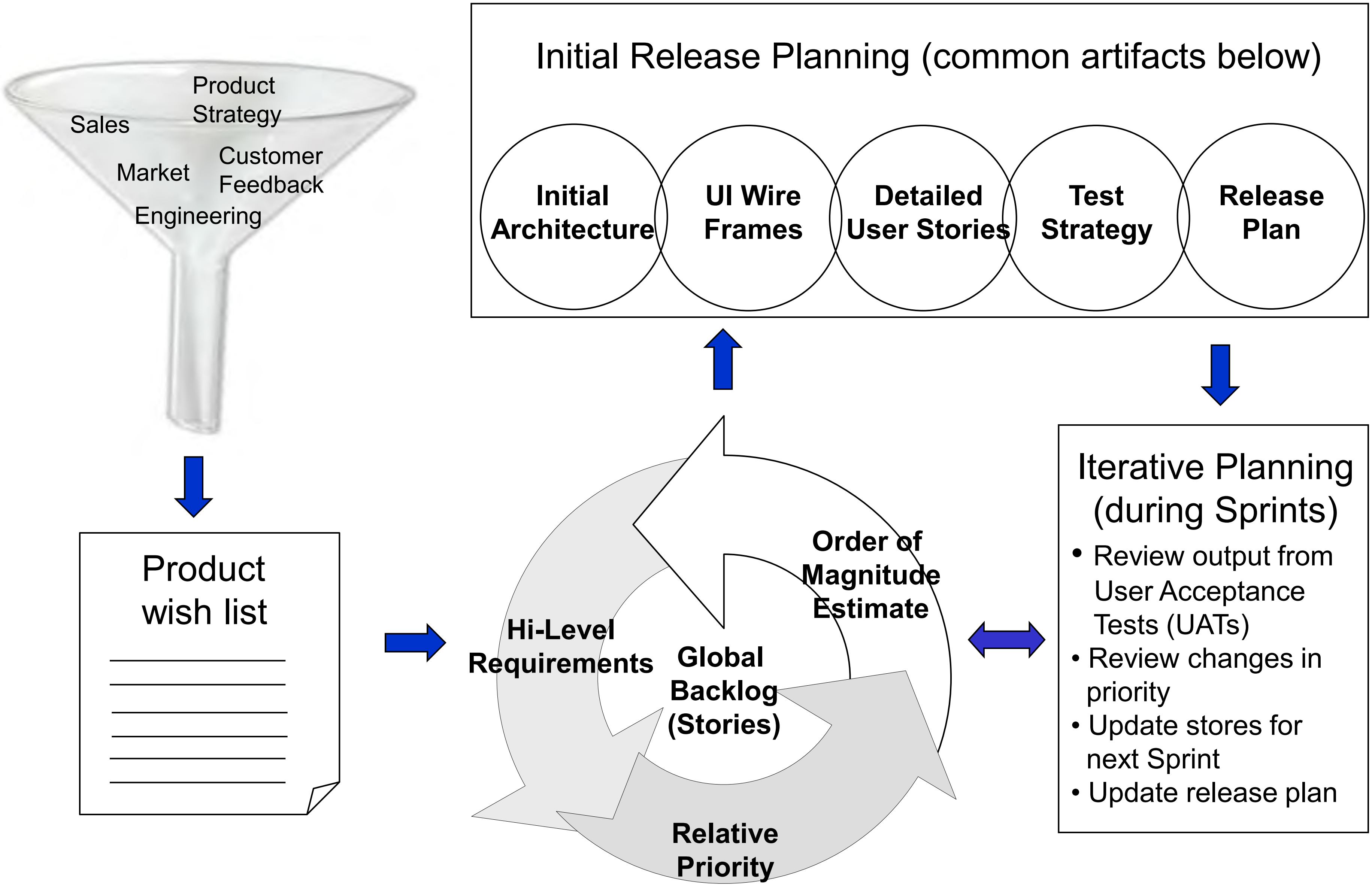


- Incremental product delivery process that encompasses all aspects of the organization
- Team-oriented with day-to-day interactions between all functions

Agile Best Practices



Agile Planning Process



Product Strategy

Sales

Customer Feedback

Market

Engineering

Product wish list

Initial Release Planning (common artifacts below)

Initial Architecture

UI Wire Frames

Detailed User Stories

Test Strategy

Release Plan

Order of Magnitude Estimate

Hi-Level Requirements

Global Backlog (Stories)

Relative Priority

Iterative Planning (during Sprints)

- Review output from User Acceptance Tests (UATs)
- Review changes in priority
- Update stores for next Sprint
- Update release plan

Roles of Core Planning Team Members

- *Business Stakeholder(s)* – Own the product vision and helps make sure the requirements meet the needs of the end customer
- *Project Manager* – Responsible for the end-to-end planning process
- *Lead Architect* – Responsible for the initial architecture and scoping
- *Lead Business Analyst* – Acts as SME and documents requirements
- *QA Lead* – Responsible for overall test strategy
- *Web Designer* – <u>Optional</u> depending upon whether UI prototyping is involved

Others participate in the process as required and necessary

Creating a Product Wish List

- A ‗Wish List‘ is a list of all possible features & functions that a particular product might encompass over time
- Inputs into the Wish List should come from everywhere within and outside of the organization that has a stake in the product
- Organization‘s often encompass a Wish List within a Product Vision document
- Product management typically owns the Wish List

Creating a Global Backlog

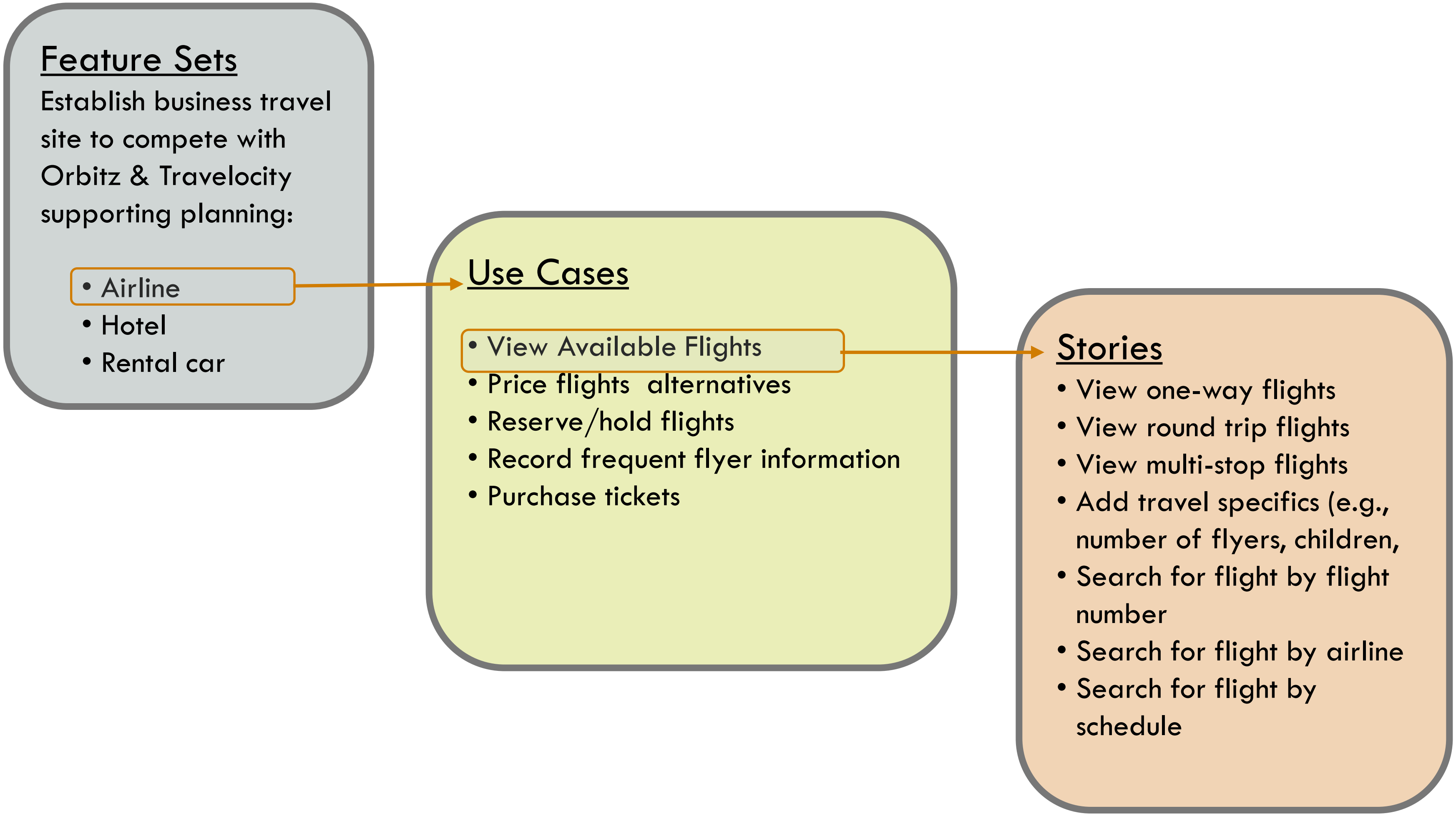
- A Global Backlog encompasses all items from the Wish List that Product Management deems the highest priority for inclusion in upcoming releases.
- Backlog items are prioritized and assigned an Order of Magnitude (OOM) estimate
- OOM can be used for budgeting purposes BUT ONLY IF THE TOP END VARIANCE ESTIMATE IS USED
- A Global Backlog contains User Stories

User Stories

- A story represents some small slice of visible, usable functionality—typically something a user can do with the system
- A well-written story possesses the following characteristics:
  - Understandable
  - Testable
  - Valuable to the customer
  - Independent of each other
  - Small enough to build a handful each Sprint
- Stories are written during initial planning or during a Sprint planning meeting once the project has begun.
- Although the idea for a story will most likely originate from the business stakeholders, many team members may have a hand in authoring the story card, including project managers, tech leads, analysts, and testers.

Creating a Global Backlog

Transforming Feature Sets into User Stories



Feature Sets

Establish business travel site to compete with Orbitz & Travelocity supporting planning:

- Airline
- Hotel
- Rental car

Use Cases

- View Available Flights
- Price flights alternatives
- Reserve/hold flights
- Record frequent flyer information
- Purchase tickets

Stories

- View one-way flights
- View round trip flights
- View multi-stop flights
- Add travel specifics (e.g., number of flyers, children,
- Search for flight by flight number
- Search for flight by airline
- Search for flight by schedule

Initial Release Planning

- Planning within Agile is iterative
- Regardless, some planning must be done up front for a variety of reasons:
  - Build a release plan the organization can plan around
  - Resolve upfront architectural tradeoffs so implementation conforms to an overall architectural vision
  - Prototype / wireframe a UI to get early feedback from stakeholders on the requirements
  - Prepare development & testing platforms accordingly
- Detail out initial Sprint(s) and projected releases for more detailed budgeting purposes

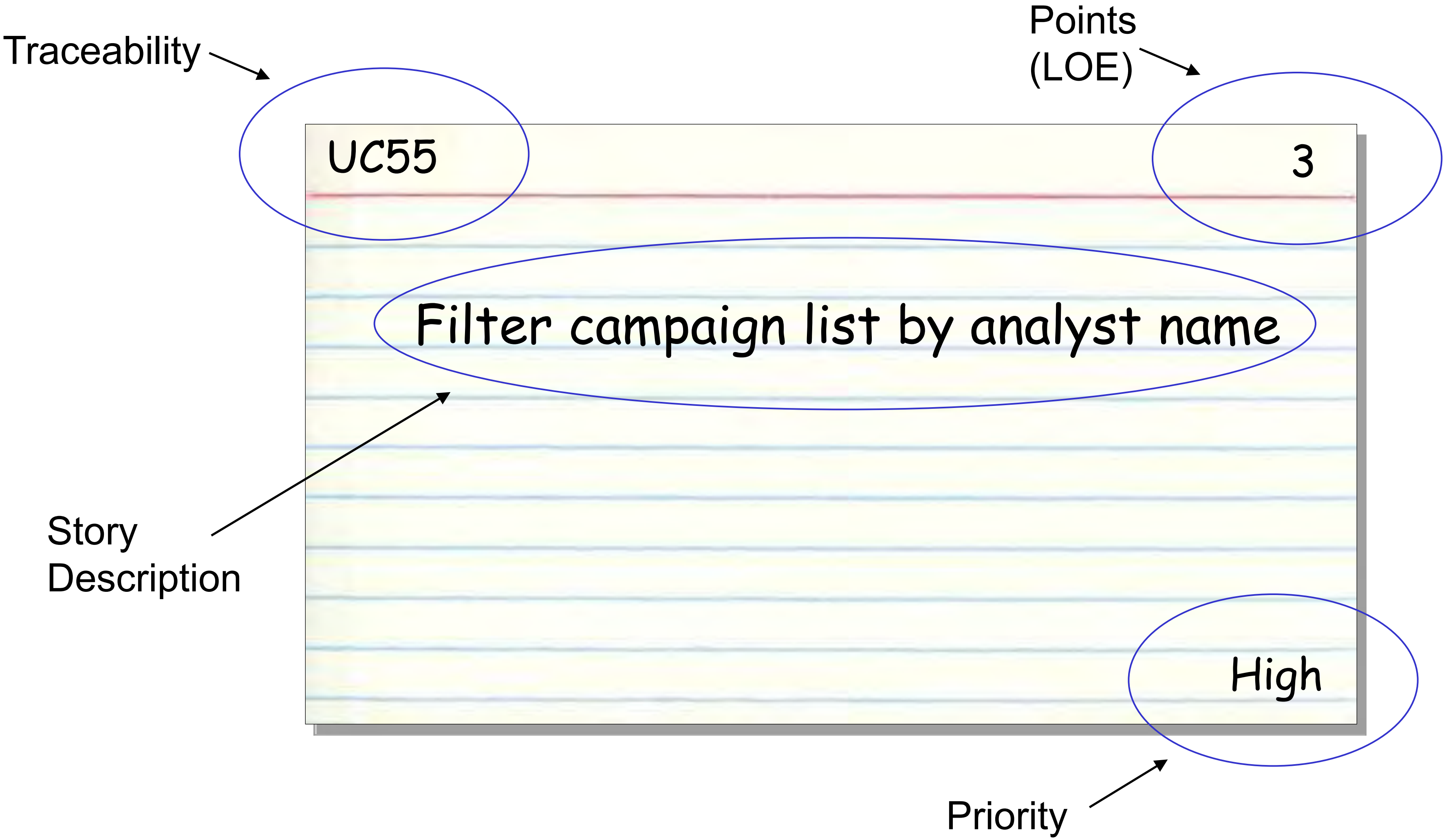
Initial Release Planning

Am I Ready to Begin?

- Is there a clear understanding of the reasons that the desired software is being developed?
- Is there an understanding of constraints under which the delivery team will have to work?
- Do the product owners have a clear vision of the desired software down to the theme or module level?
- Are the product owners and/or stakeholders identified and given full authority to make decisions on the tactical direction of the software to be developed?
- Are dependencies on people, processes or systems well understood?

Initial Release Planning

Detailed User Stories



Traceability

Points (LOE)

UC55

3

Filter campaign list by analyst name

High

Story Description

Priority

Helpful tips for user stories

1. Think in terms of inputs & outputs
   - What data does a person/system put in?
   - What data comes out?
   - **How will we test this?**
2. Think in terms of vertical slices
   - What is a minimal version of the desired functionality?
   - Can you exercise multiple layers of the system?
   - Be careful with —user views …” stories
3. Don‘t worry too much about getting it right
   - It‘s OK to rip up a card and start over
   - You will get many chances to look at a story

Example User Stories

- Which are good stories and which are not so good?

X User can use webmail.
☑ User views a message list with no messages.
☑ User views all their messages.
X System uses Log4J to log all error messages.
X Graphing and charting shall be done using Business Objects.
☑ User configures the number of messages displayed on the page.
☑ User exports their resume to Microsoft Word.
X Develop the persistence framework.
X Develop the resume view JSP.

Story Estimation using Points

- A point is a unit of measurement that is used to communicate the level of effort of a user story.
- A point is equal to one day of development/test time for a single developer/tester
- As every developer is different (level of experience, skill set, etc.), you must assign points based upon the —average" throughput of your team
- Velocity is the total points that can be complete in one Sprint

Determine Velocity for Sprints

1. Determine the Maximum Velocity (MV)
   - MV = (Sprint duration – 2) x number of developers
   - Sprints are reduced by two days to account for Kickoff and UAT
2. Determine the Realistic Maximum Velocity (RMV)
   - RMV = MV * velocity multiple
   - Velocity multiple accounts for hours spent on overhead, reviews, vacation plans, all-hands, staff meetings, etc.
   - Velocity multiple is typically between .5 and .8 depending upon the organization

- Determine Velocity for Sprints
  - Assume a ramp-up as teams get acclimated
  - Typically use 50% of RMV for first Sprint Velocity, 75% of RMV for second Sprint Velocity, and 100% of RMV for all remaining Sprints

Velocity Calculation Example

- Assumptions
  - Two week Sprints
  - 4 Developers
  - High overhead organization
- Maximum Velocity = (10 – 2) x 4 = 32 Points
- Realistic Maximum Velocity = 32 x .5 = 16 Points
- Sprint Velocity
  - 1st Sprint = 16 x 0.5 = 8 Points
  - 2nd Sprint = 16 x .75 = 12 Points
  - 3rd Sprint = 16 x 1.0 = 16 Points

Building a Release Plan

- Release plans that include Sprints* are built using:
  - Prioritized stories that include Points estimates
  - Team size
  - A decision around Sprint duration
  - A decision around how much functionality is enough to justify a release

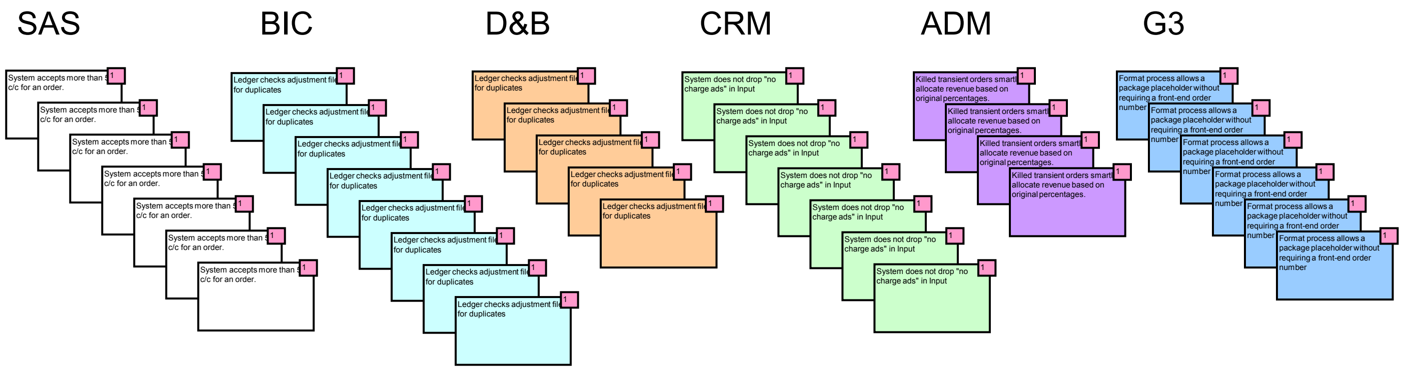
- Sprint duration
  - Tradeoff between cost of change and organization‘s agility
  - Typically 2 – 4 weeks in duration

- Release decision
  - Tradeoff between cost of deployment/release and market dynamics
  - Releases vary from daily to 3 months (huge variation!)
  - Releases are now a business decision

**A Sprint is a development iteration in SCRUM terminology*

Building a Release Plan

Story “Backlog”



SAS

System accepts more than 5 c/c for an order. 1

System accepts more than 5 c/c for an order. 1

System accepts more than 5 c/c for an order. 1

System accepts more than 5 c/c for an order. 1

System accepts more than 5 c/c for an order. 1

System accepts more than 5 c/c for an order. 1

System accepts more than 5 c/c for an order. 1

BIC

Ledger checks adjustment file for duplicates 1

Ledger checks adjustment file for duplicates 1

Ledger checks adjustment file for duplicates 1

Ledger checks adjustment file for duplicates 1

Ledger checks adjustment file for duplicates 1

Ledger checks adjustment file for duplicates 1

Ledger checks adjustment file for duplicates 1

Ledger checks adjustment file for duplicates 1

D&B

Ledger checks adjustment file for duplicates 1

Ledger checks adjustment file for duplicates 1

Ledger checks adjustment file for duplicates 1

Ledger checks adjustment file for duplicates 1

Ledger checks adjustment file for duplicates 1

CRM

System does not drop "no charge ads" in Input 1

System does not drop "no charge ads" in Input 1

System does not drop "no charge ads" in Input 1

System does not drop "no charge ads" in Input 1

System does not drop "no charge ads" in Input 1

System does not drop "no charge ads" in Input 1

System does not drop "no charge ads" in Input 1

ADM

Killed transient orders smartly allocate revenue based on original percentages. 1

Killed transient orders smartly allocate revenue based on original percentages. 1

Killed transient orders smartly allocate revenue based on original percentages. 1

Killed transient orders smartly allocate revenue based on original percentages. 1

G3

Format process allows a package placeholder without requiring a front-end order number 1

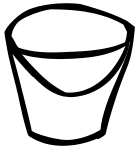
Format process allows a package placeholder without requiring a front-end order number 1

Format process allows a package placeholder without requiring a front-end order number 1

Format process allows a package placeholder without requiring a front-end order number 1

Format process allows a package placeholder without requiring a front-end order number 1

Format process allows a package placeholder without requiring a front-end order number 1

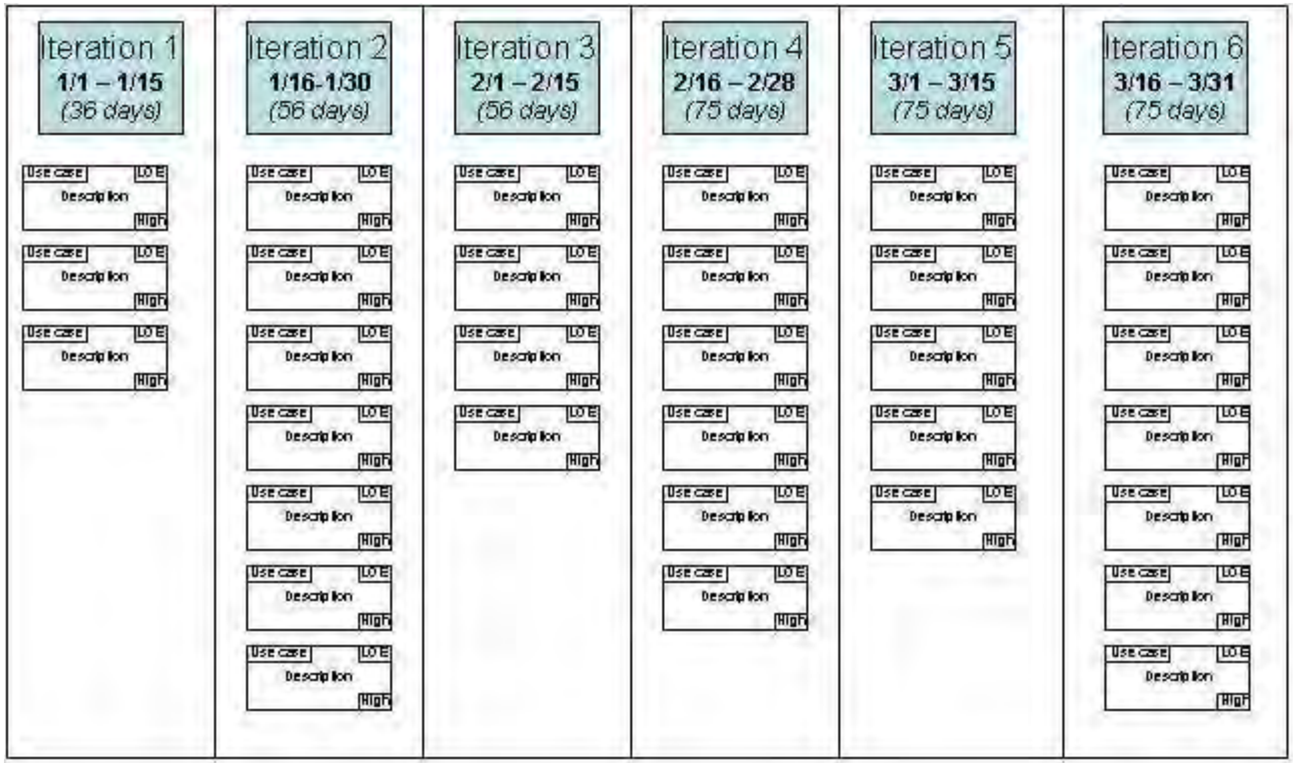


= team velocity

	Sprint 302	*Sprint 303*	*Sprint 304*
SAS	System accepts more than 5 c/c for an order. 1 System accepts more than 5 c/c for an order. 1 System accepts more than 5 c/c for an order. 1		
BIC	Ledger checks adjustment file for duplicates 1 Ledger checks adjustment file for duplicates 1 Ledger checks adjustment file for duplicates 1 Ledger checks adjustment file for duplicates 1		
D&B			
CRM	System does not drop "no charge ads" in Input 1 System does not drop "no charge ads" in Input 1		
ADM			
G3	Format process allows a package placeholder without requiring a front-end order number 1 Format process allows a package placeholder without requiring a front-end order number 1 Format process allows a package placeholder without requiring a front-end order number 1		
Total Points	**42**	**42**	**42**

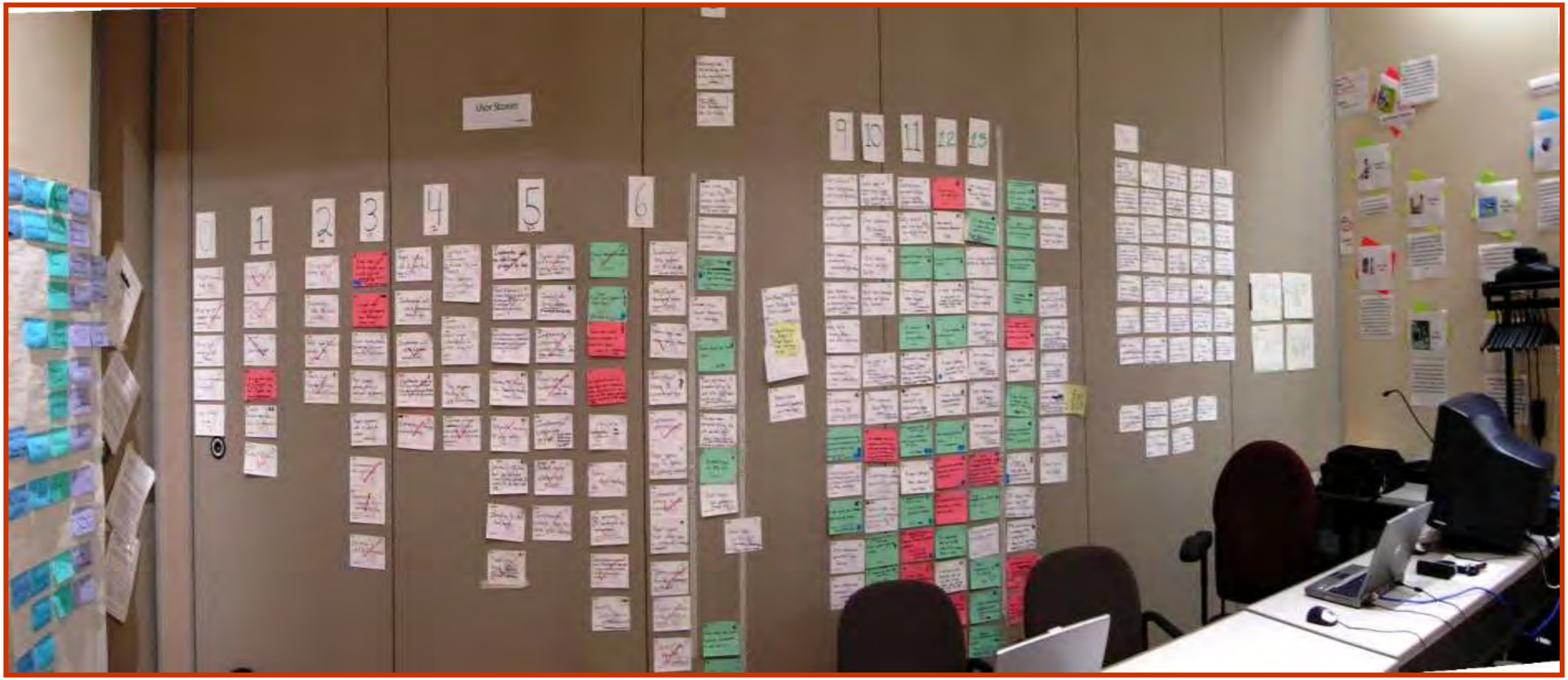
Building a Release Plan

User Stories by Sprint



Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6
1/1 – 1/15	**1/16-1/30**	**2/1 – 2/15**	**2/16 – 2/28**	**3/1 – 3/15**	**3/16 – 3/31**
(36 days)	*(56 days)*	*(56 days)*	*(75 days)*	*(75 days)*	*(75 days)*
Use case LOE Description High	Use case LOE Description High	Use case LOE Description High	Use case LOE Description High	Use case LOE Description High	Use case LOE Description High
Use case LOE Description High	Use case LOE Description High	Use case LOE Description High	Use case LOE Description High	Use case LOE Description High	Use case LOE Description High
Use case LOE Description High	Use case LOE Description High	Use case LOE Description High	Use case LOE Description High	Use case LOE Description High	Use case LOE Description High
	Use case LOE Description High	Use case LOE Description High	Use case LOE Description High	Use case LOE Description High	Use case LOE Description High
	Use case LOE Description High		Use case LOE Description High	Use case LOE Description High	Use case LOE Description High
	Use case LOE Description High		Use case LOE Description High		Use case LOE Description High
	Use case LOE Description High				Use case LOE Description High

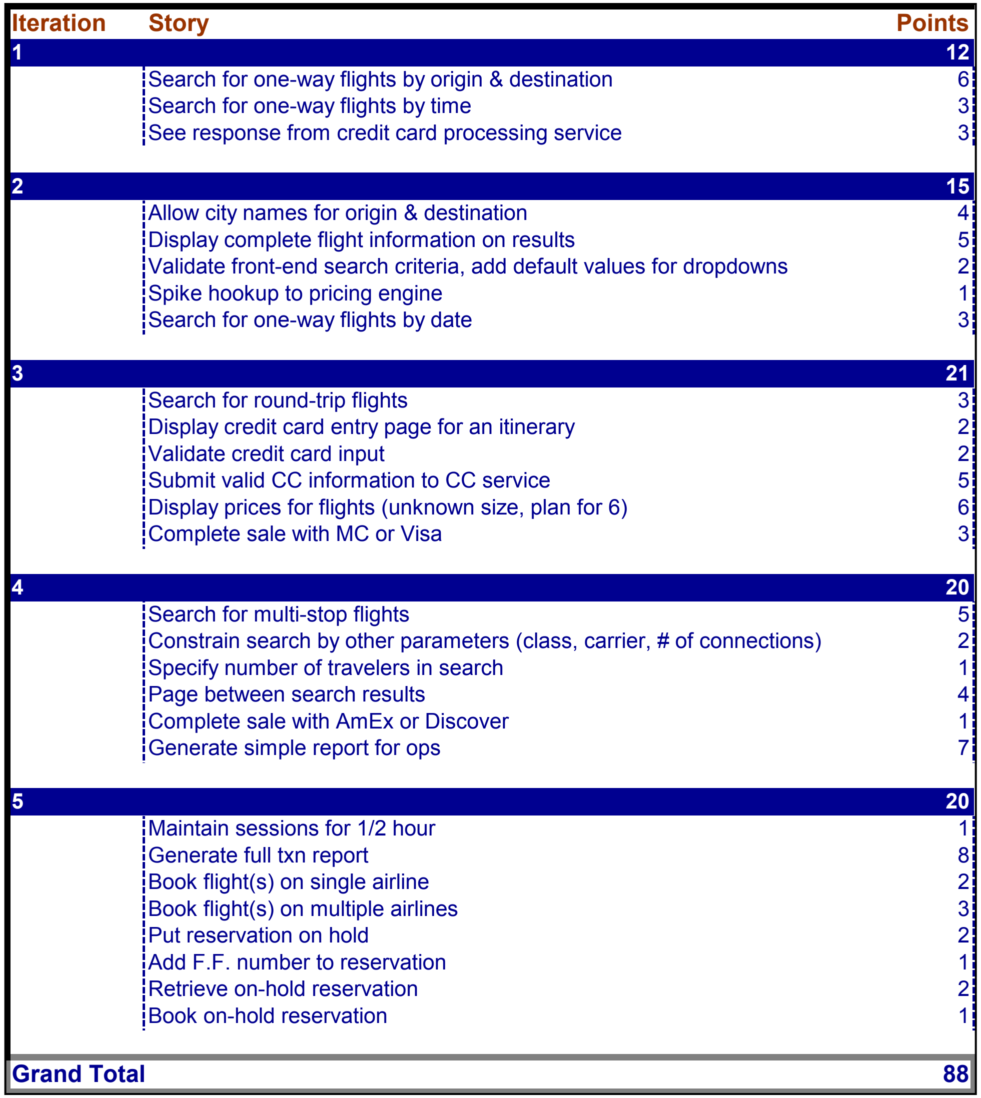
The Team Room Approach to Release Plan Mgmt



Pros – very visible and tangible, great for co-located teams, easy to modify
Cons – not under version control, harder for distributed teams to visualize, takes space

Building a Release Plan

The Virtual Approach to Release Plan Management



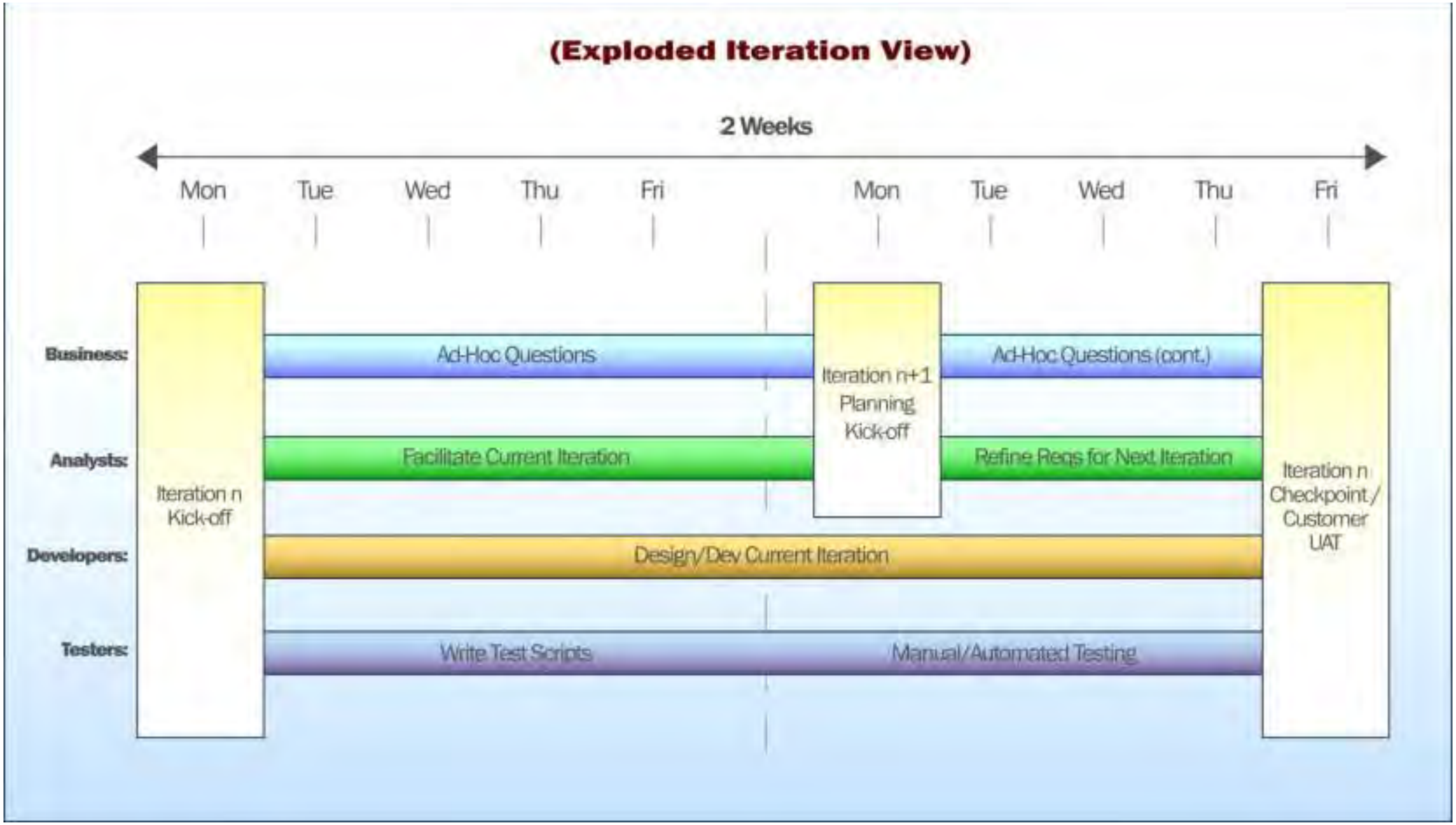
Iteration	Story	Points
1		12
	Search for one-way flights by origin & destination	6
	Search for one-way flights by time	3
	See response from credit card processing service	3
2		15
	Allow city names for origin & destination	4
	Display complete flight information on results	5
	Validate front-end search criteria, add default values for dropdowns	2
	Spike hookup to pricing engine	1
	Search for one-way flights by date	3
3		21
	Search for round-trip flights	3
	Display credit card entry page for an itinerary	2
	Validate credit card input	2
	Submit valid CC information to CC service	5
	Display prices for flights (unknown size, plan for 6)	6
	Complete sale with MC or Visa	3
4		20
	Search for multi-stop flights	5
	Constrain search by other parameters (class, carrier, # of connections)	2
	Specify number of travelers in search	1
	Page between search results	4
	Complete sale with AmEx or Discover	1
	Generate simple report for ops	7
5		20
	Maintain sessions for 1/2 hour	1
	Generate full txn report	8
	Book flight(s) on single airline	2
	Book flight(s) on multiple airlines	3
	Put reservation on hold	2
	Add F.F. number to reservation	1
	Retrieve on-hold reservation	2
	Book on-hold reservation	1
Grand Total		88

Building a Release Plan

Detail Initial Sprint(s)

- Build detailed requirements from User Stories for initial Sprint(s)
  - Typically captured in a requirements document
  - Traceable to User Stories
- Build a test plan for testing requirements and user stories
  - Define test cases and scripts
  - Test requirements and end-to-end scenarios

Iterative Development Process


(Exploded Iteration View)
2 Weeks
Mon
Tue
Wed
Thu
Fri
Mon
Tue
Wed
Thu
Fri
Business:
Analysts:
Developers:
Testers:
Iteration n Kick-off
Ad-Hoc Questions
Facilitate Current Iteration
Iteration n+1 Planning Kick-off
Ad-Hoc Questions (cont.)
Refine Reqs for Next Iteration
Design/Dev Current Iteration
Write Test Scripts
Manual/Automated Testing
Iteration n Checkpoint / Customer UAT

Typical Roles

- Project Manager – responsible for the day to day functional delivery of the software, managing project schedule and priorities, and working with stakeholders to resolve any project issues
- Architect – responsible for coding, design and architecture standards review and compliance, solutions definition and overall performance characteristic of the software
- Analyst – supports project manager in the proper definition of requirements
- Development Lead – responsible for day to day technical implementation of the software and technical management of developers
- Developers – responsible for technical implementation of the software
- QA/Test Lead – responsible for the day to day testing, verification and validation of the software, compliance, management of the testers and automation of the test cases
- Testers – responsible for the testing, verification and validation of the software and the automation of the test cases
- Business stakeholder or proxy – available when needed to answer questions regarding the product, market, customer needs

Team Communication during Agile Development

- Effective communication between all team members is absolutely critical to a successful Agile project

- A meeting rhythm should be established to assure communication happens at least at key Sprint junctures

- Important team meetings include:
  - Sprint kickoffs
  - Daily standups
  - User acceptance testing
  - Retrospectives

Wrap-Up

Agile books we recommend

- Beck, Kent, —Extreme Programming – Embracing Change”, Addison-Wesley Professional, 2004
- Cohn, Rob, —User Stories Applied”, Addison-Wesley Professional, 2004
- Cohn, Rob, —Agile Estimating & Planning”, Prentice Hall PTR, 2005
- Crispin, Lisa, —Agile Testing – A Practical Guide for Testers & Agile Teams”, Addison-Wesley Professional, 2009
- Duvall, Paul, —Continuous Integration: Improving Software Quality and Reducing Risk”, Addison-Wesley Professional, 2007

Questions?

- Contact information:
  - Jeffery Payne, Coveros, Inc.
  - 703-431-2920
  - jeff.payne@coveros.com